African Virtual University

Applied Computer Science: CSI 2100

FUNDAMENTALS OF COMPUTER ORGANISATION AND ARCHITECTURE

Ashenafi Kassahun

Foreword

The African Virtual University (AVU) is proud to participate in increasing access to education in African countries through the production of quality learning materials. We are also proud to contribute to global knowledge as our Open Educational Resources are mostly accessed from outside the African continent.

This module was developed as part of a diploma and degree program in Applied Computer Science, in collaboration with 18 African partner institutions from 16 countries. A total of 156 modules were developed or translated to ensure availability in English, French and Portuguese. These modules have also been made available as open education resources (OER) on oer.avu.org.

On behalf of the African Virtual University and our patron, our partner institutions, the African Development Bank, I invite you to use this module in your institution, for your own education, to share it as widely as possible and to participate actively in the AVU communities of practice of your interest. We are committed to be on the frontline of developing and sharing Open Educational Resources.

The African Virtual University (AVU) is a Pan African Intergovernmental Organization established by charter with the mandate of significantly increasing access to quality higher education and training through the innovative use of information communication technologies. A Charter, establishing the AVU as an Intergovernmental Organization, has been signed so far by nineteen (19) African Governments - Kenya, Senegal, Mauritania, Mali, Cote d'Ivoire, Tanzania, Mozambique, Democratic Republic of Congo, Benin, Ghana, Republic of Guinea, Burkina Faso, Niger, South Sudan, Sudan, The Gambia, Guinea-Bissau, Ethiopia and Cape Verde.

The following institutions participated in the Applied Computer Science Program: (1) Université d'Abomey Calavi in Benin; (2) Université de Ougagadougou in Burkina Faso; (3) Université Lumière de Bujumbura in Burundi; (4) Université de Douala in Cameroon; (5) Université de Nouakchott in Mauritania; (6) Université Gaston Berger in Senegal; (7) Université des Sciences, des Techniques et Technologies de Bamako in Mali (8) Ghana Institute of Management and Public Administration; (9) Kwame Nkrumah University of Science and Technology in Ghana; (10) Kenyatta University in Kenya; (11) Egerton University in Kenya; (12) Addis Ababa University in Ethiopia (13) University of Rwanda; (14) University of Dar es Salaam in Tanzania; (15) Universite Abdou Moumouni de Niamey in Niger; (16) Université Cheikh Anta Diop in Senegal; (17) Universidade Pedagógica in Mozambique; and (18) The University of the Gambia in The Gambia.

Bakary Diallo

The Rector

African Virtual University

Production Credits

Author

Ashenafi Kassahun

Peer Reviewer

Dessalegn Mequanint

AVU - Academic Coordination

Dr. Marilena Cabral

Overall Coordinator Applied Computer Science Program

Prof Tim Mwololo Waema

Module Coordinator

Victor Odumuyiwa

Instructional Designers

Elizabeth Mbasu

Benta Ochola

Diana Tuel

Media Team

Sidney McGregor

Barry Savala

Edwin Kiprono

Kelvin Muriithi

Victor Oluoch Otieno

Michal Abigael Koyier

Mercy Tabi Ojwang

Josiah Mutsogu

Kefa Murimi

Gerisson Mulongo

Copyright Notice

Supported By

AVU Multinational Project II funded by the African Development Bank.

Table of Contents

Course Overview

I wish to welcome you to this course, known as, Fundamentals of Computer Organization and Architecture. This is a module where you will learn about the computer components, their functions and how they are interconnected. In the module, the functions and the design of the various computer functional units will be discussed. These concepts will help students to get some idea about Computer Organization and Architecture. The course will basically be dealing with the operational units of the computer and how they are interconnected using their architectural specifications.

The prerequisites

Introduction to Computer Science

Materials

The materials required to complete this course units for this module are:

1. Computer
2. Logic chip boards
3. Logic gates
4. System Softwares(Operating Systems)
5. Secondary Storage Devices

Reading materials and other resources

1. Computer Essentials by Isaac Hurst Price: Free! Words: 6,020. Language: English. Published: March 13, 2013. Categories- [choose the option appropriate to you].

2. www.bookrix.com/_ebook-a-m-williams-computer-literacy- computer Literacy by A.M Willikams- Read free book on line.

3. http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471467413.html

4. http://www.srmuniv.ac.in/downloads/computer_architecture.pdf- computer architecture

5. http://read.pudn.com/downloads166/ebook/758721/Solution.pdf

6. http://www.scs.carleton.ca/sivarama/org_book/org_book_web/slides/chap_1_versions/ch1_1.pdf

7. https://www.rose-hulman.edu/Users/faculty/young/CS-Classes/csse232/review/CODF_v02b.pdf

Course Goals

At the end of this course, the learner should be able to

1. Illustrate and describe basic functional operations components a computer.

2. Identify the structure and the interconnection structures of the computer

3. Demonstrate a good understanding of the basics of numbering systems

4. Identify and analyze the gate symbols for the Boolean operations AND, OR, NOT and XOR.

5.Explain how the control unit of the Central processing unit interprets machine level instructions.

Units Overview

Unit 0: Unit Introduction

This unit is basically to be used for diagnostic purposes. You will use it to remind yourself of those concepts that you learnt in computer applications including the hardware (e.g mouse, keyboard, screen, printers, CPU, scanners) and software (e.g Ms office application software, Windows, Unix, Linux) that are used in different application areas.

Unit 1:System Architecture and Design

The system architecture and design is an important unit of the course where you will learn about the various components of the computer such as the central processing unit (CPU), the memory, input and output units (I/O) and other storage devices. The functions of these components and where they are located within the computer system will be discussed.

Unit 2: Data Representation and Logic

In this unit, the basics of the numbering system will be discussed. These are the decimal, octal, hexadecimal and the binary numbering system. You will also learn how to convert from one system to another and how data is represented in each of those notations. The computer building units will be discussed based on the logic gates which use the binary boolean expressions. The unit will also introduce the gate symbols for Boolean operations. It is expected that at the end of the unit, you will be able to identify, describe and analyze the gate symbols for the Boolean operations AND, OR, NOT and XOR. These are the basics of computer design. You will also be introduced to the construction of a truth-table with the input/output behavior of each individual gate.

Unit 3: Micro architecture (Computer Organization)

In this unit, you will learn about the organization of the computer, the part of computer architecture that defines the data paths, data processing and storage elements, as well as how these elements should be implemented in the Instruction Set Architecture (ISA) discussed in the unit 4. The implementation is in the form of computer instructions. Computer instructions are composed of an operation code, also known as opcode, and one or more operands. An operand is something which is acted on by the operation. It can be a constant value, a register identification code, or the address of a location in memory or of an input /output device.

Unit 4: Instruction Set Architecture (ISA)

In this unit, you will learn about the Instruction Set Architecture which is the part of the central processing unit that is visible to the programmer who writes the compiler instructions. Each type of central processing unit is designed to understand a specific group of instructions called the instruction set. This set has an architecture known as instruction set architecture (ISA). An instruction set architecture (ISA) is the interface between the computer's software and hardware and also can be viewed as the programmer's view of the machine. It defines the codes that a central processor reads and acts upon. An interface is a shared boundary across which two separate components of a computer system exchange information.

Assessment

Formative assessments, used to check learner progress, are included in each unit.

Summative assessments, such as final tests and assignments, are provided at the end of all units and cover knowledge and skills from the entire module. Summative assessments are administered at the discretion of the institution offering the course. The suggested assessment plan is as follows:

1	Unit assignment of varying complexity and weights	20%
2	Mid Term exam	10%
3	Final exam	70%
Total		100

Schedule

Unit	Unit title	Activities	Estimated time
0	Unit Introduction	Activity 0.1-Check your understanding (Looking for materials to answer the quiz for this unit and making verifications)	3 hours
1	System Architecture and Design	Reading listed resource list to understand the features and location of computer components • Input unit devices • The motherboard • The bus • Output unit devices	10 hours
		Internet search for additional materials on the computer components	5 hours
		Labworks- opening of computers to prepare for identification of components activities including Image Examination in the listed materials and comparing with what is in the opened computers	10 hours
		Activity 1.1- Identification of input devices	4 hours
		Activity 1.2 - Identification of components connecting to the motherboard Activity 1.3 -Identification of output unit devices)	1 hour
		Activity 1.4 - Identification of components of the the bus	5 hour
		e) Formative assessment	1 hour

2	Data Representation and Logic	Readings listed materials to explore basics numbering systems,understand how CPU's control unit interprets a machine-level instruction, identify and analyze different logic gate symbols for the Boolean operations.	10 hours
		Lab works: Design logic gates, determine the outputs for the following. • AND gate • OR gate • NOT gate • NAND gate • XOR gate • EXOR gate	10 hours
		Activity 2.1 -Details Number system conversion Activity 2.2 -Construction of Logic gates for the respective labworks above Activity 2.3-Construction of truth tables for the respective labworks	2 hours 3 hours 3 hours
		d) Internet search for additional reading materials on number system and boolean operations	7 hours
		Formative unit assessments (C.A.T)	2 hours
3	Micro architecture (Computer Organization)	Reading the listed resources list and the materials for the unit Internet search	7 hours 7 hours
		Activity 3.1- Interconnection structures- The Bus	5 hours
		Activity 3.2:- Reflection Activity Instruction cycle(The Fetch-Execute Cycle)	5 hours
		Activity 3.3 -Instruction cycle(The Fetch-Execute Cycle)	5 hours
		e) Formative assessment (C.A.T)	2 hours
4		Reading the listed resources list and the materials for the unit	8 hours

	Instruction Set Architecture (ISA)	Activity 4.1 -Direct addressing mode	1 hour
		Activity 4.2-Indirect addressing mode	1 hour
		Activity 4.3- Reflective	1 hour
		Assessment (C.A.T)	2 hours
		TOTAL	120 Hours

Units Readings Materials and Other Resources

The readings and other resources in this course are:

Unit 0:Unit Introduction

Pre- assessment

Required readings and other resources:

1. http://www.tutorialspoint.com/computer_fundamentals/index.htm- Computer Fundamentals

Unit 1 System Architecture and Design

Pre- assessment

1.http://www.karbosguide.com/books/pcarchitecture/start.htm- PC Architecture a book by Michael Karbo- chapter 2 to chapter 8.

2.http://simple.wikipedia.org/wiki/Computer_architecture - computer architecture

3.http://en.wikiversity.org/wiki/Computer_architecture_and_organization - Computer architecture and organization.

4.http://www.tutorialspoint.com/computer_fundamentals/index.htm- Computer Fundamentals-(Read from computer types to computer hardware)

5.http://en.wikipedia.org/wiki/Computer_architecture#Instruction_set_architecture- computer architecture.

Unit 2:-Data Representation and Logic

Pre- assessment

Required readings and other resources:

1.http://www.karbosguide.com/books/pcarchitecture/start.htm- PC Architecture a book

by Michael Karbo.

2. http://www.tutorialspoint.com/computer_fundamentals/index.htm- Computer Fundamentals-computer Fundamentals

3.http://en.wikipedia.org/wiki/Computer_architecture#Instruction_set_architecture- computer architecture

4.http://www.edupub.gov.lk/Administrator/English/10/ICT%20g-10%20E%20new%20syllbus/Chapter%203.pdf- Data Representation methods in a computer system.

5.http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/index.html #notgate: Basic Gates and Functions

Unit 3 Micro architecture (Computer Organization)

Required readings and other resources:

1. http://en.wikibooks.org/wiki/A-level_Computing/AQA/Computer_Components,_The_Stored_Program_Concept_and_the_Internet/Machine_Level_Architecture/Machine_code_and_processor_instruction_set- Machine Level Architecture: Machine code and processor instruction set

2. http://en.wikibooks.org/wiki/A-level_Computing/AQA/Computer_Components,_The_Stored_Program_Concept_and_the_Internet/Machine_Level_Architecture/Machine_code_and_processor_instruction_set- Machine Level Architecture: Machine code and processor instruction set

3. http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm : Logic Gates

Unit 4. Instruction Set Architecture (ISA)

Unit readings and other resources

The readings in this unit are to be found at course level readings and other resources.

1. http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Mips/format.html Instruction format

2. http://en.wikipedia.org/wiki/MIPS_instruction_set MIPS instruction set

3. http://en.wikipedia.org/wiki/Computer_architecture computer architecture

4.http://simple.wikipedia.org/wiki/Computer_architecture - computer architecture

5. http://en.wikiversity.org/wiki/Computer_architecture_and_organization - Computer architecture and organization.http://en.wikipedia.org/wiki/Computer_architecture.

6. http://www.karbosguide.com/books/pcarchitecture/start.htm PC Architecture a book by Michael Karbo. chapter 29.

7 http://en.wikipedia.org/wiki/Multiprocessing multiprocessing.

8. http://homepage.cs.uri.edu/book/cpu_memory/http://cs.sru.edu/~mullins/cpsc100book/IntroToCS.htmlcpu_memory.htm How Computers Work: The CPU and Memory.

9. http://www.tutorialspoint.com/computer_fundamentals/index.htm- Computer Fundamentals

10. http://en.wikipedia.org/wiki/Computer_architecture#Instruction_set_architecture computer architecture.

11. http://www.cs.toronto.edu/~demke/469F.06/Lectures/Lecture6.pdf- Lecture on Interrupts (University of Toronto)

12. http://people.freebsd.org/~jhb/papers/bsdcan/2007/article/article.html PCI Interrupts for x86 Machines under FreeBSD

13. http://en.wikipedia.org/wiki/Instruction_set Instruction set

Unit 0. Unit Introduction

Unit Introduction

The purpose of this unit is to determine your understanding of previous knowledge related to this course . In this unit, you will be given assessment that is geared towards reminding your of self the common usage of the computer functions and capabilities which you have encountered before. You will explore the common parts of the computer and what they are used for. This is to prepare you for more detailed discussion which you will learn about these parts and their functions in the other units.

Unit Objectives

Upon completion of this unit you should be able to:

1. Identify the main components of a computer.
2. Describe the functions main components of computer identified above..

Key Terms

Hardware :The physical components of the computer system.

Software: The programs or instructions that tell the computer what to do.

CPU: The brain of the computer or central processing unit.

ROM: The permanent memory that is built in your computer. This is read only.

RAM: The computer's working memory, sometimes called random-accessed memory.

Megabyte: Approximately a million bytes.

Gigabyte: Approximately a billion bytes (or 1,000 megabytes).

Input Device: The hardware that is used to pass information into the computer.

Output Device: The hardware that receives and displays information coming from the computer.

Modem; The device that allows your computer to talk to other computers over a telephone line.

Monitor; A video or computer display device.

Laser Printer; A printer that uses both laser and photographic technology to produce high quality output.

Printer; The hardware that provides printed output from the computer.

Hard Copy; A printed copy of computer output.

Compact Disc; A disc on which a laser has digitally recorded information such as audio, video, or computer data.

Hard Disk; A fixed, large-capacity magnetic storage medium for computer data.

Floppy Disk; A portable magnetic storage medium for computer data that allows users to randomly access information.

Graphical User Interface; The use of graphical symbols instead of text commands to control common computer functions such as copying programs and disks.

Icon; A small picture or symbol representing a computer hardware function or component.

Ink-jet Printer; A type of printer that forms letters on the page by shooting tiny electrically charged droplets of ink.

Unit Assessment

Check your understanding

Identify the parts of the computer system as shown in figure 1

Figure 1.0 : Parts of Computer System

(Retrieved from http://www.proprofs.com/quiz-school/story.php?title=label-computer-input-output-parts- Label The Computer, Input, Output, Parts)

1. Which component or the hardware part is responsible for arithmetic/logic operations?

    a: CPU

    b: RAM

    c: OS

2. Apart from the computer screen, name two other devices that are used to display the output from a computer

3. What is a computer system unit?

4. What is the name of the internal component that all other components tie into?

5. Which one is an INPUT device?

    a. Floppy Disk

    b. Keyboard

    c. Monitor

    d. Speakers

6. Which one is a STORAGE device?

a. CPU

b. Headphones

c. Floppy Disk

d. Modem

Grading Scheme

No weight/mark is allotted for the assessment questions.

Feedback

Follow the link and see if you have answered correctly.

1. http://www.proprofs.com/quiz-school/story.php?title=label-computer-input-output-parts

Unit Readings and Other Resources

The readings in this unit are to be found at the course-level section "Readings and Other Resources".

1. https://drive.google.com/viewerng/

=ZGVmYXVsdGRvbWFpbnxjYXNpdGVmb3JzdHVkZW50c3xneDoxOWY2NTE2NWE4MTk4MDcx - Course Objectives - Computer Organization and Architecture

2. http://en.wikibooks.org/wiki/Introduction_to_Computer_Information_Systems/Application_Software- Introduction to Computer Information Systems/Application Software

Unit 1: System Architecture and Design

Introduction

This unit will include the components of a computer including input, central processing unit (CPU), the memory, storage and output. The functions of these components will also be covered including where they are found in a computer. The unit will introduce you to the basic principles of computer architecture. These principles will include how the computer motherboard is able to provide connections to all other components to form a computer system.

Unit Objectives

At the end of this unit, you should be able to:-

1. Identify the functional components of a computer.
2. Describe the functions of the identified computer components, and
3. List the main principles of computer architecture.

Key Terms

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system, (Systems architecture, Retrieved April 27, 2016, from https://en.wikipedia.org/wiki/Systems_architecture)

Computer architecture is a specification detailing how a set of software and hardware technology standards interact to form a computer system or platform. In short, computer architecture refers to how a computer system is designed and what technologies it is compatible with (Computer_architecture , Retrieved April 27, 2016, from https://en.wikipedia.org/wiki/Computer_architecture)

Introduction

Functional Components of Computer System

Computers have input devices such as keyboard, mouse,joystick, light pen, trackball, scanner graphic tablet, microphone,magnetic ink card reader (MICR), optical character reader(OCR), bar code reader, optical mark reader(OMR). Computers also have output devices that are used to bring to the computer what the computer stores. These output devices include monitors, graphic plotters and printers (Computer fundamentals Tutorials, Retrieved from http://www.tutorialspoint.com/computer_fundamentals/index.htm)

The diagram showing the relationships between the input units, processing units and the output as is shown in the diagram below.

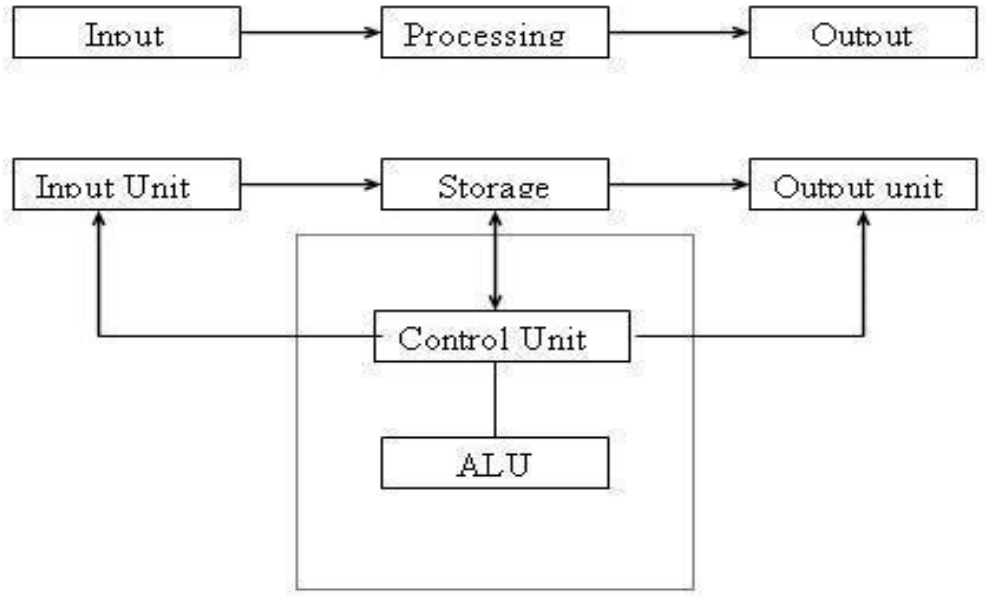


Figure 1.2: Functional Components of Computer System (Von Neumann Architecture)

Retrieved on 7th March, (2016) Available from wiki educator free learning content at: (http://wikieducator.org/File:CR-rie.jpg)

Input Unit

Input units is made up of those devices that translate data into a form the computer can understand. They are divided into three types which are Keyboard hardware, the Pointing device and the Source data-entry.

Keyboard hardware:This is the part of the input unit that converts letters, numbers, and other characters into electrical signals that are machine-readable by the computer's processors(retrieved from https://www.koofers.com/flashcards/is-chapter-5/review).It looks like typewriter keyboard and contains alphabetical & alphanumeric characters, numbers and other function keys.

Pointing devices: This part controls the position of the cursor or pointer on the screen. Example are; mice, light-pens, touchpad's ,joysticks (adapted from http://www.knowsh.com/NotesSearch/NotesDetail/80105/Input-Devices,-Keyboard,-Mouse,-Joy-Stick,-Light-pen,-Track-Ball,-Scanner,-Graphic-Tablet,-Microphone,-Magnetic-Ink-Card-Reader,-Optical-Character-Reader,-Bar-Code-Reader,-Optical-Mark-Reader)

Source data-entry devices: these are devices that are not keyboards or pointing devices. Data-entry devices that create machine-readable data on magnetic media or paper or feed it directly into the computer's processor, without the use of a keyboard. Categories include scanning devices (imaging systems, bar-code readers, mark and character – recognition devices, and fax machines), audio-input devices, video input, photographic input (digital cameras), voice-recognition systems, sensors, radio-frequency identification devices, and human-biology-input devices (https://www.koofers.com/flashcards/is-chapter-5/review)

Storage Unit

This unit is made up of those devices that are to store data or instructions in the computer system. They divided into three types (a) main memory (b) secondary memory and (c) the registers (Computer fundamentals Tutorials, Retrieved from http://www.tutorialspoint.com/computer_fundamentals/index.htm)

Main memory: The main memory is used for holding data and instructions which will be needed for use by the central processing unit. It's characterized by fast accessing information, low capacity and high costs. They are two main types

1. RAM – Random Access Memory

They can both be read, to retrieve information or written into, to store information. The contents of RAM remain stable as long as power is available i.e. volatile and has a short time response.

Figure 1. 3: Computer memory

(Computer fundamentals Tutorials, Retrieved from http://www.tutorialspoint.com/computer_fundamentals/index.htm)

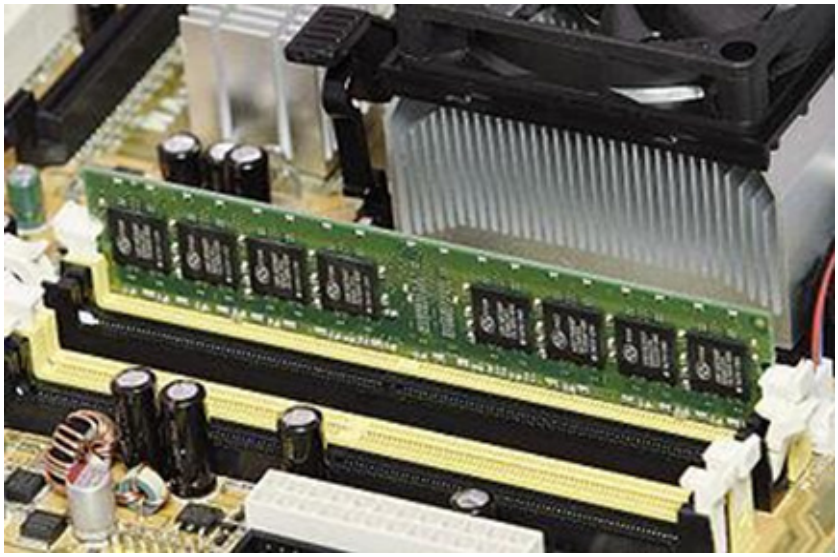


Figure 1.4: Computer System Memory Slot

Source: https://en.wikibooks.org/wiki/How_To_Assemble_A_Desktop_PC/Assembly

a. Static RAM (SRAM)

b. Dynamic RAM (DRAM)

Static RAM (SRAM) retain its contents even as long as the power is available. Used by manufactures to programme computer system programs. For example programming basic input output systems (BIOS) programs. Once the SRAM is written into the contents cannot be erased. It changes to ROM

(Computer fundamentals Tutorials, Retrieved from http://www.tutorialspoint.com/computer_fundamentals/computer_ram.htm and Computer organization Tutorial, Retrieved April 27, 2016, from http://www.tutorialspoint.com/computer_logical_organization/memory_devices.htm

Figure 1. 5 : RAM Module

Available and Retrieved on 12th March, (2016) at

http://www.tutorialspoint.com/computer_fundamentals/computer_ram.htm

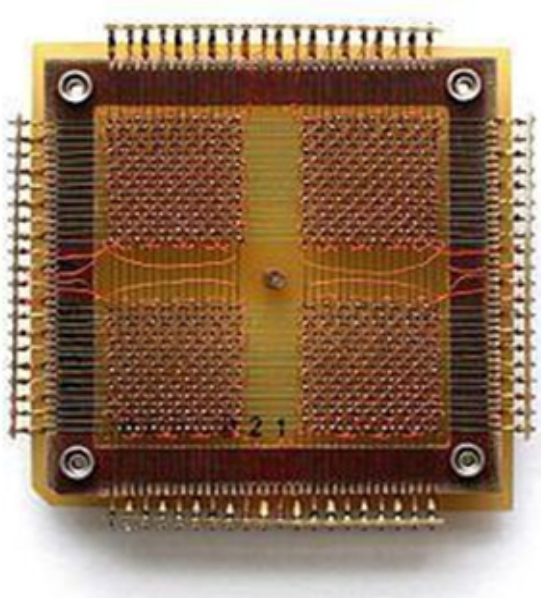


Figure 1.6: A 32 x 32 core memory plane storing 1024 bits of data.

Available and Retrieved on 12th March, (2016) at: :http://en.wikipedia.org/wiki/Magnetic-core_memory

Characteristic of the SRAM (Learn computer fundamentals. (2016). Retrieved March 14, 2016, from http://www.tutorialspoint.com/computer_fundamentals/computer_ram.htm-)

Dynamic RAM (DRAM)

DRAM, must be continually refreshed (connected to power) in order to maintain the data. This is done connecting the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is commonly used by most system memory because it is small and cheap. All DRAMs are made of memory cells which are composed of one transistor and one capacitor..

Characteristics of the Dynamic RAM (Learn computer fundamentals. (2016). Retrieved March 14, 2016, from http://www.tutorialspoint.com/computer_fundamentals/computer_ram.htm)

The Dynamic RAM has a short data lifetime, it has to be refreshed often, is known to be slower than SRAM. It is smaller in size and can be used as RAM. DRAM is expensive and consumes less power.

Read Only Memory (ROM)

This memory provides a permanent or a semi permanent storage. Its contents can be read but it cannot be rewritten within the normal computer operations. The memory is non volatile and is not affected by power outages. The memory stores instructions that are used to start a computer in what is known as a bootstrap operation.

There are various types of ROM (available at Learn computer fundamentals. (2016) , Retrieved March 14, 2016, from http://www.tutorialspoint.com/computer_fundamentals/computer_ram.htm). They include

1. Masked ROM (MROM) considered to be the first ROMs that were hard-wired and contained preprogrammed set of data or instructions. They are not very expensive.
2. Programmable Read only Memory (PROM)

This a read only memory that can be modified by a user only once by entering desired contents using PROM program.It can be programmed once and is not erasable.

(available at Learn computer fundamentals. (2016). Retrieved March 14, 2016, from http://www.tutorialspoint.com/computer_fundamentals/computer_ram.htm) -

Advantages of ROM

1. Non-volatile in nature
2. These cannot be accidentally changed
3. Cheaper than RAMs
4. Easy to test
5. More reliable than RAMs
6. These are static and do not require refreshing
7. Its contents are always known and can be verified

(Computer organization Tutorial, Retrieved April 27, 2016, from http://www.tutorialspoint.com/computer_logical_organization/memory_devices.htm and in Computer fundamentals Tutorials, Retrieved from http://www.tutorialspoint.com/computer_fundamentals/computer_ram.htm)

Storage standards

Storage standards are used for storing backup information that is not needed immediately by the CPU. They are characterized by slow access of information, higher capacity and lower cost. Examples: hard disk, floppy, CD, DVD .

Sector and track in hard disk

A Sector in the context of computing refers to a small area of a storage device, for example a hard disk drive. A typical hard disk drive when low-level formatted is split into tracks, sectors and clusters:

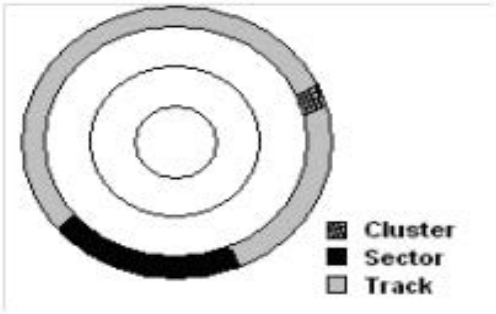


_Figure 1.7: Cluster, sector and track

Retrieved March 14, 2016, Available at : http://www.helpwithpcs.com/jargon/sector-track-cluster.htm

CD-ROM and DVD-ROM

DVD is optical disc storage media format used for data storage. DVDs are of the same form factor as compact discs (CDs), but can store more data than CDs

Advantages of Storage standards

1. Storage capacity: Going by the storage capacity, the DVD specifications have four disk configurations, ranging from 4.7 GB to 17 GB. It ranges from single-sided and single-layer disks.

2. Speed of data transfer: The data transfer rate of DVD ROM is approximately equivalent to a 92X CD ROM.

3. Reliability: CD ROMs are very reliable and have a long shelf life yet if compared to the DVD, the DVD scores over the CDs in terms of reliability. Since DVD ROM discs are made of plastics bonded together, the discs are more rigid than CD ROM discs.

(Computer fundamentals Tutorials, Retrieved from http://www.tutorialspoint.com/computer_fundamentals/index.htm)

There are two types of access storage devices methods

1. Sequential Access Storage Devices (SASD): Data is stored in sequential order. Retrieval is also sequential.

Characteristics of SASD

a. Storage media is magnetic tape.

b. Supports batch processing environment

c. Excellent form of backup Computer organization Tutorial, Retrieved April 27, 2016, from http://www.tutorialspoint.com/computer_logical_organization/memory_devices.htm)

2. Direct Access Storage Devices (DASD): Data can be stored and retrieved randomly.

Characteristics of DASD

a.Storage capacity has high density. How tightly packed data is on the disk.

b.DASD is required for transaction processing.(Computer organization Tutorial, Retrieved April 27, 2016, from http://www.tutorialspoint.com/computer_logical_organization/memory_devices.htm)

Registers

Registers are high speed circuits that are a staging area for temporarily storing data during processing. This is a part of Central Processor Unit, so they reside inside the CPU. The information from main memory is brought to CPU and keep the information in register. Due to space and cost constraints, we have got a limited number of registers in a CPU. These are basically faster devices.

Cache Memory

This is a high speed memory which is used to speed up the CPU. It is found in between the CPU and the main memory. Its main purpose is to hold the data and programs which are frequently used by the CPU. These data and programs are transferred from the disk to this memory by the operating system and the CPU can then access them. This memory also stores data that may service future requests for the same data and makes the access faster. What is stored may be results of an earlier computation and/or the duplicates of data that is stored elsewhere.The use of this memory has certain advantages that include the following.

1. It is faster than main memory.
2. It consumes less access time as compared to main memory.
3. It stores the program that can be executed within a short period of time.
4. It stores data for temporary use. There are also a list of its disadvantages that include:-

The disadvantages of cache memory are as follows:

1. It has limited capacity.
2. It is volatile and is very expensive.

Computer fundamentals Tutorials, Retrieved from http://www.tutorialspoint.com/computer_fundamentals/index.htm)

Virtual Memory

Virtual memory is an addressing scheme implemented in hardware and software that allows non-contiguous memory to be addressed as if it were contiguous. It maps memory addresses used by a program, called virtual addresses, into physical addresses in computer memory. Main storage as seen by a process or task appears as a contiguous address space or collection of contiguous segments.

Programs can address data that does not currently reside in main memory. When this occurs, the hardware and operating system automatically load the data at the requested address from auxiliary storage into main memory. This occurs transparently to the user program. As a result, programs can reference a larger amount of (RAM) memory than actually exists in the computer using the stored concept facilitate by system software in managing main memory and secondary memory to swap contents between the two storage systems location.

System Board [Motherboard]

The motherboard contains the CPU, the memory and all the connectors to the other parts of the computer system. It provides connection for all the other components (adapted from Parts of a motherboard. Retrieved May 3, 2016, from http://study.com/academy/lesson/what-is-a-motherboard-definition-function-diagram.html)

A typical motherboard has:-

1. A CPU socket- this is where the CPU is directly connected. The power socket has a number of heat sinks and there are also mounting points where fans that ensure that the CPU does not overheat. When in use, the CPU generates a lot of heat. Heat.

2. There is a power connector which distributes power to the central processing unit and the other components of the motherboard.

3. The motherboard has a chip that acts as an interface between the central processing unit, the memory and the other components of the motherboard. This chip is known as the Northbridge. This generates lots of heat and has a large heat sink. There is also another chip that is used to control the input/output functions (I/O). It is known as the southbridge. Unlike the northbridge, this chip is not connected to the CPU. The two chips are combined to to form a single chipset as shown in figure 1.11.

4. There are several connectors that provide physical interfaces between the I/O devices and the motherboard. The Southbridge is the one that handles these connections.

5. There are integrated Drive Electronics (IDE) and the Serial Advance Technology Advancement (SATA) which are slots for hard drives used to store files.

6. There is also a ROM chip that contains the startup instructions for the system. This chip is also known as the BIOS.

7. There are slots for videos and graphics cards like the Accelerated Graphics Port (AGP) and the Peripheral Component Interconnect Express (PCIe)

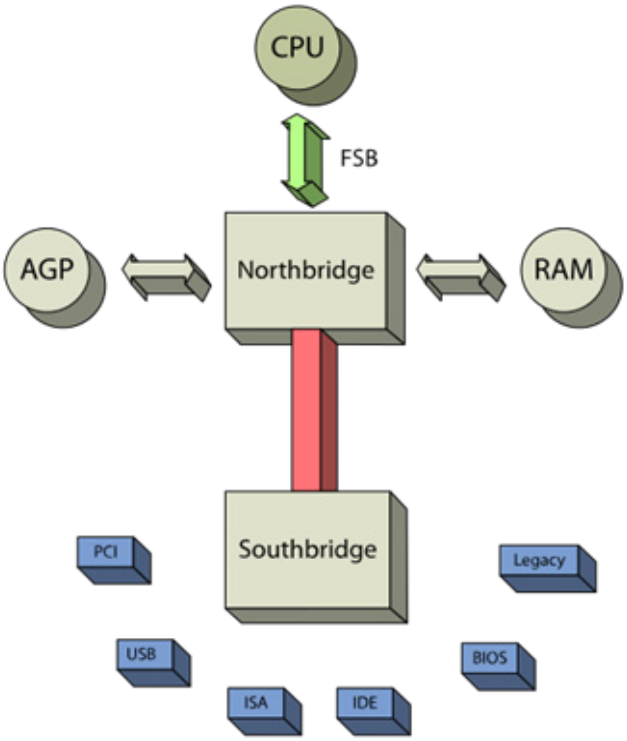


Figure: 1.11 "Chipset schematic" by Fred the Oyster.

Avaible and retrieved on 12th March, (2016) at: :http://commons.wikimedia.org/wiki/File:Chipset_schematic.svg#/media/File:Chipset_schematic.svg

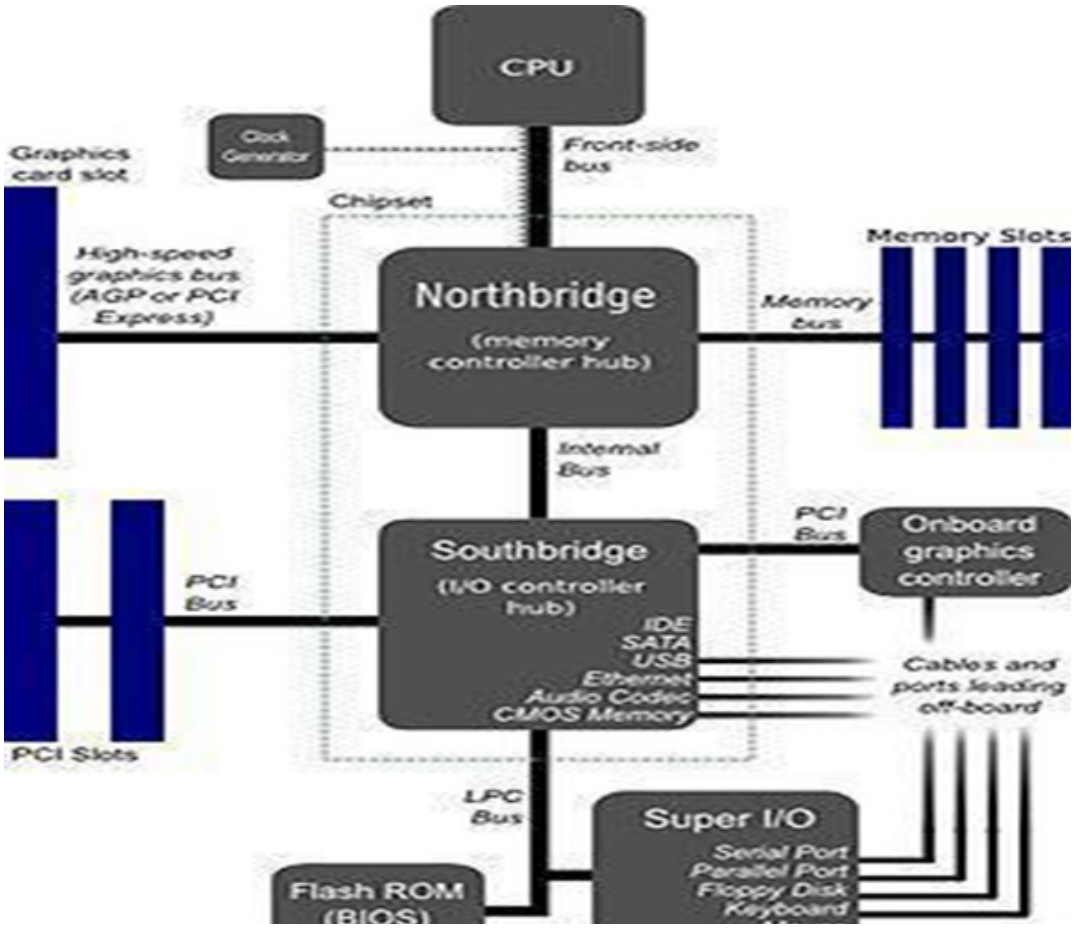


Figure: 1.12: Moxfyre (Image:Motherboard_diagram.svg).

Available & retrieved on 12th March, (2016) at: P. M. (n.d.). Introduction to computers. Retrieved May 3, 2016, from http://cs.sru.edu/~mullins/cpsc100book/module03_internalHardware/module03-01_internalHardware.html

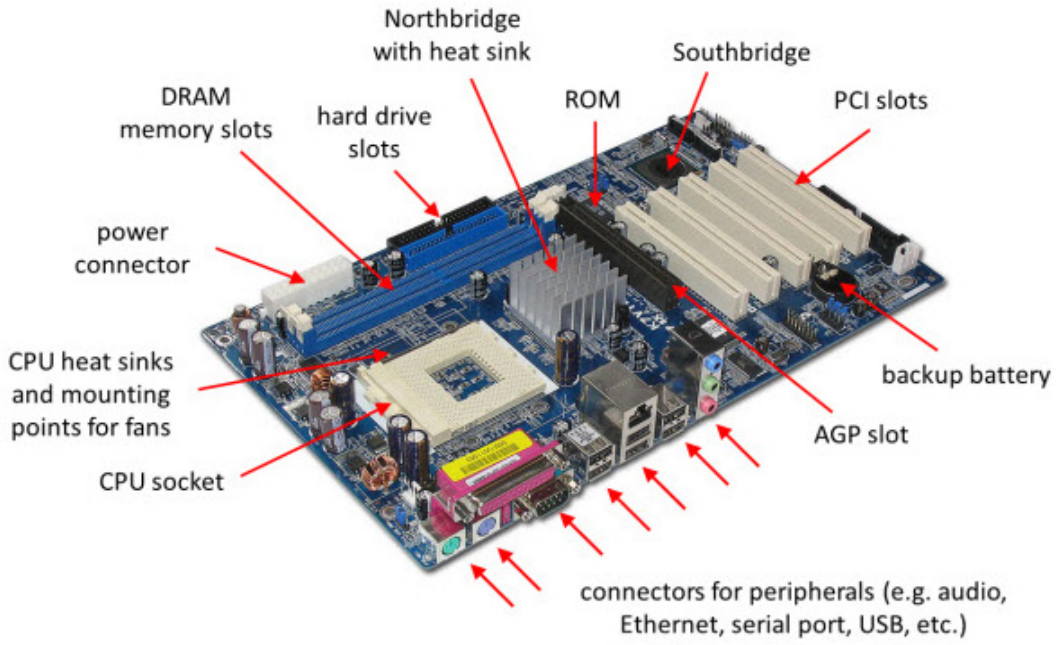


Figure 1.13: System board main components and parts.

(adapted from Parts of a motherboard. Retrieved May 3, 2016, from http://study.com/academy/lesson/what-is-a-motherboard-definition-function-diagram.html)

An example of PCI devices is shown in the diagrams that follows:

Figure 1.14a: A typical 32-bit, 5 V-only PCI card, in this case, a SCSI adapter from Adaptec._

(adopted from PCI card adapter. (n.d.). Retrieved May 3, 2016, from https://commons.wikimedia.org/wiki/File:32-bit_PCI_card.JPG

Motherboards for different models are of various designs as shown in the diagrams that follow.

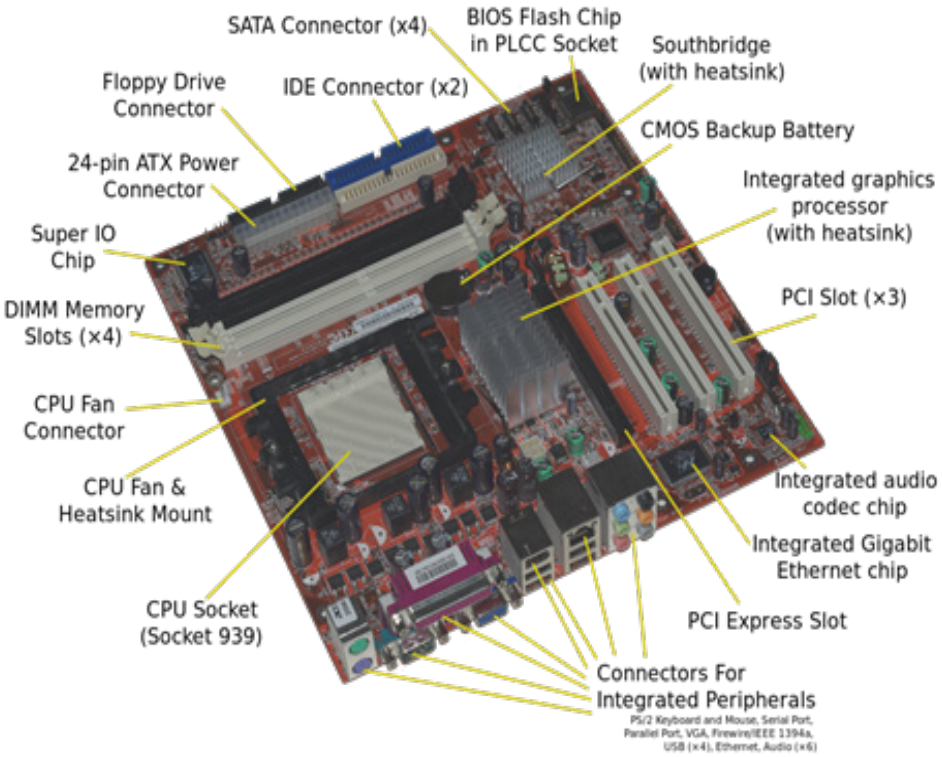


Figure 1.14c :Acer E360 Socket 939 motherboard by Foxconn.

Acer Socket Motherboard. (Retrieved May 3, 2016, from https://commons.wikimedia.org/wiki/File:Acer_E360_Socket_939_motherboard_by_Foxconn.svg

Creative Commons Attribution-Share Alike 3.0 Unported)

Motherboard for an Acer desktop personal computer, showing the typical components and interfaces that are found on a motherboard. This model was made by Foxconn in 2007, and follows the ATX layout (known as the "form factor") usually employed for desktop computers. It is designed to work with AMD's Athlon 64 processor

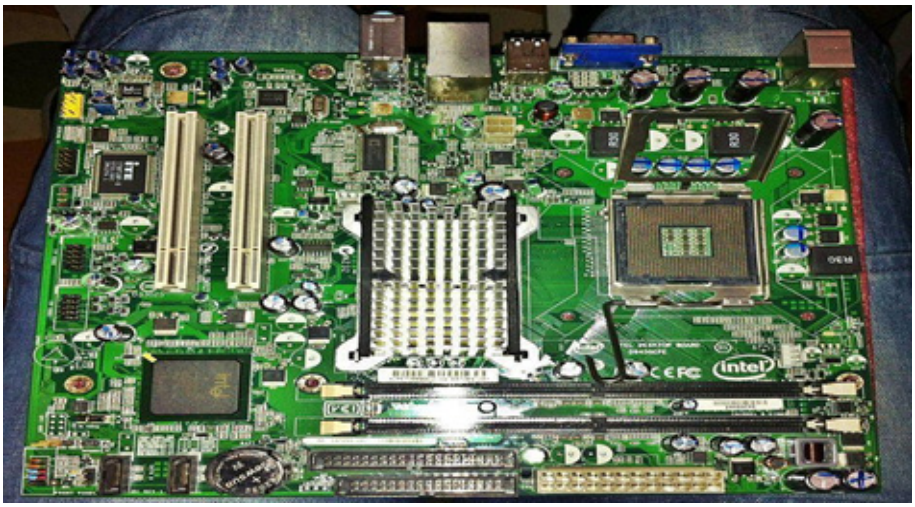


Figure:1.14d: Intel D945GCPE A microATX Motherboard LGA775 for Intel Pentium 4, D, XE, Dual-Core, Core 2 (circa 2007)

(Adopted from J. P. (n.d.). Intel Motherboard. Retrieved May 3, 2016, from https://commons.wikimedia.org/wiki/File:Intel_D945GCPE_Board.JPG

Creative Commons Attribution-Share Alike 4.0 International)

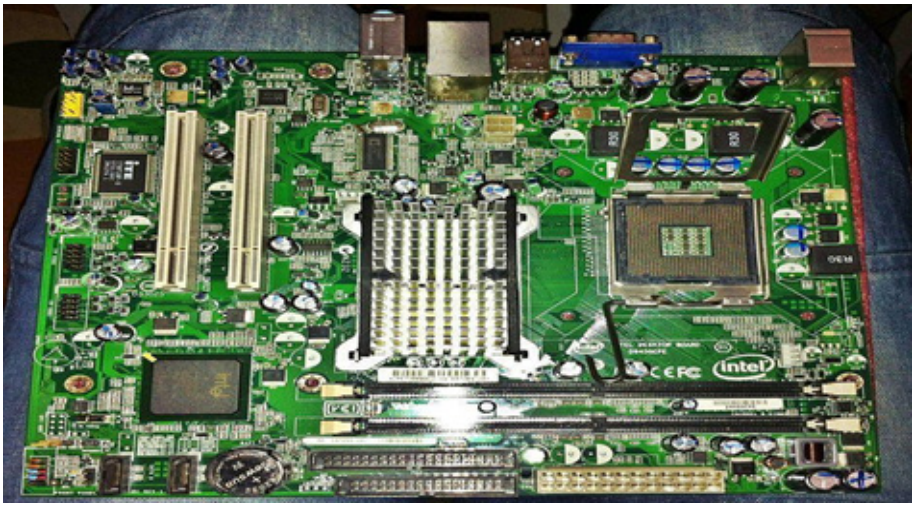


Figure 1.14e: 386DX-40 Motherboard Octek Jaguar V. 386DX40 MB Jaguar Motherboard.

(2005, November 20). Retrieved May 3, 2016, from https://commons.wikimedia.org/wiki/File:386DX40_MB_Jaguar_V.jpg)

The Octek Jaguar V motherboard from 1993.This board has few onboard peripherals, as evidenced by the 6 slots provided for ISA cards and the lack of other built-in external interface connectors"386DX40

Central Processing Unit (CPU)

This is seen as the brain of very computer and that is why every computer must have it. It main purpose is to perform data processing operations. The CPU also stores data, intermediate results and and the program (instructions). The CPU is also the part of the computer that controls the operations of all parts of the computer (Adapted from Central Processing Unit. (n.d.). Retrieved May 3, 2016, from http://www.tutorialspoint.com/computer_fundamentals/computer_cpu.htm)

The CPU in most models is as shown in the diagrams below.

Figure 1.8: "Intel 80486DX2 top".(Adopted from Intel 80486- CPU. (2005, November 09).

Retrieved May 3, 2016, from https://commons.wikimedia.org/wiki/File:Intel_80486DX2_bottom.jpg

Creative Commons Attribution-Share Alike 2.0 Generic)

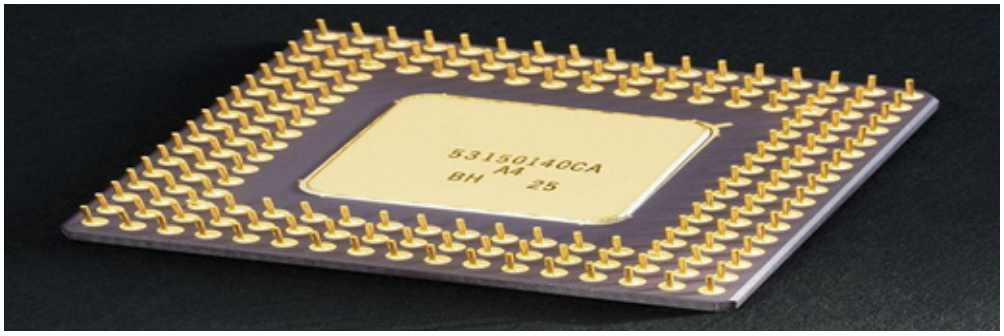


Figure 1.9: "Intel 80486DX2 bottom".

(Adopted from Intel 80486- CPU. (2005, November 09). Retrieved May 3, 2016, from https://commons.wikimedia.org/wiki/File:Intel_80486DX2_bottom.jpg

Creative Commons Attribution-Share Alike 2.0 Generic)

As mentioned earlier, the CPU being the brain of the computer, has to have all instructions pass through it to be processed.. In most literature, the CPU is simply referred to as the processor.

A typical CPU has the following components as shown in Figure 1.10.

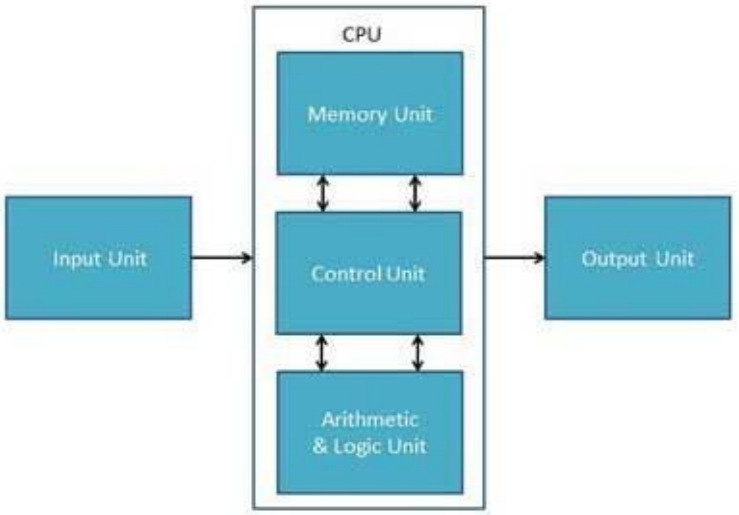


Figure 1.10 The Central Processing Unit

Source: http://www.tutorialspoint.com/computer_fundamentals/computer_cpu.htm

1.The first component is the combinational logic that is composed of two parts.

- The arithmetic section- This section is the one responsible for arithmetic operations such as addition, subtraction, multiplication and division. If there are complex operations, repetitive use of these operations is done.
- The logic section that deals with logic operations like comparison, making selections and merging data. It gets as input the operands (data words to be processed), it also receives the status information from previous operations, and a code from the control unit indicating which operation to perform._

2.The second is the control unit (CU), and this is the manager of the components of the computer. This unit, though is the manager, does not carry out any processing itself. The following are its important functions ((Adapted from Central Processing Unit, Retrieved May 3, 2016, from http://www.tutorialspoint.com/computer_fundamentals/computer_cpu.htm)

- It is responsible for controlling the transfer of data and instructions among other units of a computer.
- It manages and coordinates all the units of the computer.
- It obtains the instructions from the memory, interprets them, and directs the operation of the computer.
- It communicates with Input/ Output devices for transfer of data or results from storage.
- It does not process or store data.

To control the other components, the control unit makes use of electrical signals. It communicates with both the Arithmetic Logic Unit and the memory. The unit is the one responsible for reading and interpreting instructions obtained from the memory and thereafter converts them into a series of signals that will be used to activate the other parts of the computer.

3.The third is the cache (registers), which serves as high-speed memory where instructions can be copied to and retrieved. Modern CPUs have been constructed as a single integrated unit called a microprocessor. As such, a CPU is a specific type of microprocessor. The individual components of a CPU have become so integrated that you can't even recognize them from the outside.

(Adapted from Central Processing Unit, Retrieved May 3, 2016, from http://www.tutorialspoint.com/computer_fundamentals/computer_cpu.htm)

The CPU is connected to the rest of the system through system bus. Through system bus, data or information gets transferred between the CPU and the other component of the system. The system bus may have three components (Adopted from Introduction to the CPU, Retrieved May 3, 2016, from http://nptel.ac.in/courses/Webcourse-contents/IIT- Guwahati/comp_org_arc/msword/m5_CPU_Design/1_intro_to_cpu.txt)

1.Data bus is used to transfer the data between main memory and CPU.

2.Address bus is used to access a particular memory location by putting the address of the memory location.

3.Control Bus: Control bus is used to provide the different control signal generated by CPU to different part of the system.

These interconnections will be discussed in unit 3- Computer Organization.

The CPU is about two inches by two inches in size as in figure 1.8 and figure 1.9 .

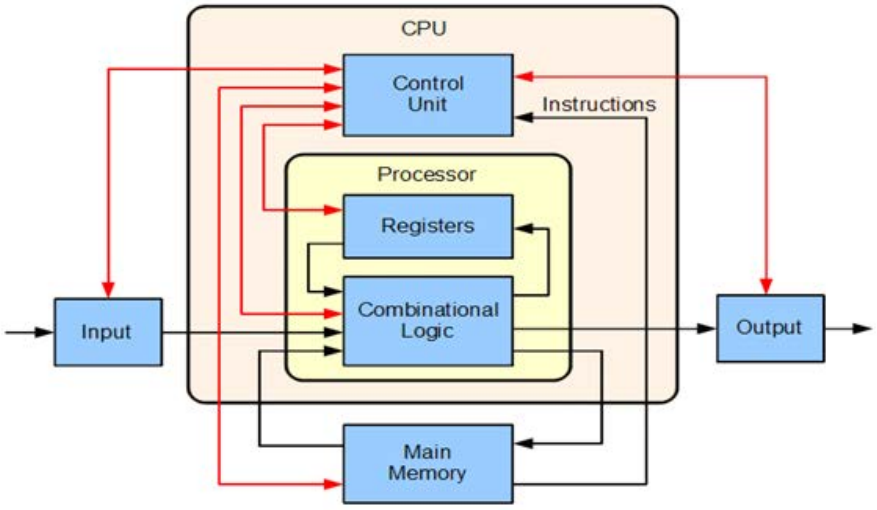


Figure 1.11: "A Basic Computer" by Lambtron

(adopted from L. (2015, January 06). A basic computer. Retrieved May 3, 2016, from https://commons.wikimedia.org/wiki/File:ABasicComputer.gif#/media/File

Creative Commons Attribution-Share Alike 4.0 International)

Learning Activities

Learning Activities experiment, reading and identification of components and the Interconnections

Before attempting these activities, read the following listed materials:-

1. http://www.tutorialspoint.com/computer_fundamentals/index.htm -Computer Components

2. http://www.contentedwriter.com/function-of-computer-hardware-components/ Functions of computer hardware.

The laboratory assistant in your lab will open for you at least two different types of computers, for example, HP compaq and Dell desktop.

Repeat the same activities using HP laptop and Dell laptop. You will be expected to document the results of each activity which will form part of your learning materials. Prepare a report which you will then send to your instructor by e-mail or shared using google drive tool.

Learning Activity 1

Activity Details :Input unit)[estimated Time 1 hour]

a. Identify the type of computers you are using.

b. Identify and describe the features of computer input devices. If the device has its own subcomponents, identify and describe their features.

c. Identify and locate the CPU and then describe its features.

d. Identify and locate the output devices and then describe them.If the device has its own subcomponents, identify and describe them.

Learning Activity 2

Activity Details: Motherboard [estimated Time 1 hour]

1. Locate and describe the features of the computer motherboard.

2. Locate the following memories on the motherboard and describe their features.

    a.Primary or main memory

    b.Secondary memory

3. Describe the roles and functions of the following in system memory management

    a. Registers

    b. Cache memory

    c. Virtual Memory

4. Locate and describe the functions of the following

    a. Control bus

    b. Data bus

    c. Address bus

Learning Activity 3

Activity Details:Output [estimated Time 1 hour]

Locate and describe the following components and their positions on the motherboard

    a. hard drives,

    b. optical drives,

    c. video card

    d. sound card

    e. Southbridge

    f. Northbridge

    g. Integrated Graphics processor

    h. Connectors for peripherals, e.g keyboard, mouse, VGA, USB, etc.

    i. Systems buses / connectors/jumps

    j. System bios

Learning Activity 4

Activity Details : The bus [estimated Time1 hour]

    a. Identify and describe the PCI of the model.

    b. Repeat the activities in 1-9 for three other models.

    c. Compare the characteristics of the different types of computers you have used in the lab and describe their differences or similarities.

Conclusion for Activities 1, 2, 3 and 4

All computers share common features, each of them has input/output units. These units have different devices, each of which perform different functions. The computer uses its input unit to get data by use of input devices. The data will then be processed using the central processing unit (CPU).. The CPU has the arithmetic logic unit and the control unit. Within the CPU there are memory structures known as the registers. The processed data will then be made available to the users outside the system by use of the output unit whose devices will then be used to display the data or make the data be available to the computer user outside the system.

Different models have different types of the devices and it is good to be able to recognize these differences.

Assessment-Computer components

A. What is an input unit?

B. What do you understand by an output unit?

C. What is the use of PCI slot?

D. What is a port?

E. What condition must be in place before you can connect a sound card to a motherboard?

F. What manages and directs the flow of data between each of the components?

Feedback

Assessment

A. What is an input unit? Answer: Inputs accepts data

B. What do you understand by an output unit? Answer:Use to display results

C. What is the use of PCI slot? Answer:Used to mount expansion

D. What is a port? Answer:Use to attach input /output

E. What condition must be in place before you can connect a sound card to a motherboard?

Answer: Compactable and sound card drivers

F. What manages and directs the flow of data between each of the components? Chip set

Unit Assessment

This unit assessment will involve a laboratory work. You will then be expected to send your answers to the instructor using his/her e-mail address or shared drive. This will be graded and the grading is as given in the grading section.

Instructions

Select one model of computer in your lab, then

a. Write down the model of the computer you have.

b. List the input/output devices it can be connected to.

c. Describe the features of the motherboard.

- How many ports does the motherboard have? List them and describe them.
- How many PCI does it have?

d. What is the speed of the processor?

Grading Scheme

This assessment is worth 8% of your total course mark as follows

1. participation and discussion- 2%

2. document - 2%

Unit Summary

In this unit, you have learnt about the components and functions of a computer. That different models have different features of some common components. Every computer has an input unit which uses various input devices to enable the computer to receive data from outside the system. There is also the output unit that uses various output devices to make processed data to be available for users of the computer to view/use them. The central processing unit is the brain of the computer and any data that gets into the computer is processed in this section. The CPU has the ALU and the control unit whose functions have been described in the unit. All these components are connected together to form a system by using a motherboard. It enables the various units to communicate with each other. The memory unit is the store of data. In the next section, you will learn about the architecture of these parts.

Unit Readings and Other Resources

The readings in this unit are to be found at course level readings and other resources.

1. http://www.tutorialspoint.com/computer_fundamentals/index.htm -Computer Components

2. http://www.contentedwriter.com/p1-explain-the-function-of-computer-hardware-components/ - Explain the functions of computer hardwa

3 .http://wikieducator.org/Caribbean_Secondary_Education_Certificate_-_Information_Technology/Functions_of_Hardware_Components_of_a_Computer_System -Caribbean Secondary Education Certificate - Information Technology/Functions of Hardware Components of a Computer System

4 .http://en.wikipedia.org/wiki/Computer_architecture- Computer architecture

5. http://www.mpsaz.org/eva/staff/ksrandle/class1/files/motherboard_labels.pdf- labeled diagram of a motherboard.

6 .http://simple.wikipedia.org/wiki/Computer_architecture - computer architecture

Unit 2:Data Representation and Logic

Unit Introduction

In this unit, you will learn the basics of numbering systems; use of decimal, octal, binary and hexadecimal number system. You will also learn how to convert standard notation expanded numbers and how to convert from one binary system to another i.e decimal, octal, binary and hexadecimal data representation.. The unit will also introduce you to how a CPU's control unit interprets a machine-level instruction – either directly or as a micro-program. You will be able to identify, describe and analyze the gate symbols for the Boolean operations AND, OR, NOT and XOR. You will be introduced to how to construct a truth-table with the input/output behavior of each individual gate.

Unit Objectives

At the end of this unit, you should be able to:-

1. Demonstrate a good understanding of basics of numbering systems
2. Use decimal, octal, binary and hexadecimal number system
3. Convert: standard notation expanded numbers
4. Identify and analyze the gate symbols for the Boolean operations AND, OR, NOT and XOR.
5. Construct a truth-table with the input/output behavior of each individual gate and construct logic gates for different boolean operations.

Key Terms

Number system:- A set of values used to represent different quantities

Decimal Number System:- This consists of ten digits from 0 to 9. These digits can be used to represent any numeric value. The base of decimal number system is 10.

Binary Number System:- This consists of two digits 0 and 1. Its base is 2. Each digit or bit in binary number system can be 0 or 1

Octal Number System:- This consists of eight digits from 0 to 7. The base of octal system is 8. Each digit position in this system represents a power of 8

Hexadecimal Number System: This consists of 16 digits from 0 to 9 and A to F. The alphabets A to F represent decimal numbers from 10 to 15. The base of this number system is 16.

Boolean Operators are simple words (AND, OR, NOT or AND NOT) used as conjunctions to combine or exclude keywords in a search, resulting in more focused and productive results.

A logic gate is an elementary building block of a digital circuit. Most logic gates have two inputs and one output. At any given moment, every terminal is in one of the two binary conditions low (0) or high (1), represented by different voltage levels.

Control unit (CU) - decodes the program instruction in the IR, selecting machine resources such as a data source register and a particular arithmetic operation, and coordinates activation of those resources.

Arithmetic logic unit (ALU) - performs mathematical and logical operations

Introduction

You will recall that a computer will require power to function. The power is either on or off. These are two states that can be represented by either "true" or "false" or "1" or "0". It can be said that the "1" and "0" are equivalent to "on" and "off" status of power into the computer.

(This is available in Representation Methods in the Computer system http://www.edupub.gov.lk/Administrator/English/10/ICT%20g-10%20E%20new%20syllbus/Chapter%203.pdf)

This computer state can then be used to represent any type of data that can be processed by a computer system. All that will be required will be how to convert that data into a system of numbers that can be understood by the computer. The series of "0's" and "1's" used by the computer is known as the binary coding system.

Number system

They are a set of values that are used to represent different quantities is known as Number System. For example, a number system can be used to represent the number of students in a class. The digital computer represents all kinds of data and information in binary numbers. It includes audio, graphics, video, text and numbers. They are represented by the computer as numbers it can understand. This group of numbers that the computer can understand is called a 'positional number system'. Some important number systems are as follows:-

a. Decimal number system

b. Binary number system

c. Octal number system

d. Hexadecimal number system

There is limited number of symbols in the number system called digits. These symbols represent different values depending on the position they occupy in the number. The value of each digit in a number can determined by examining the digit is under consideration, the position of the digit in the number and the total number of digits in the number system. The total number of digits used in a number system is called its base or radix.

Decimal Number System

This consists of ten digits from 0 to 9. These digits can be used to represent any numeric value. The base of decimal number system is 10 as it uses ten digits. The value represented by individual digit depends on weight and position of the digit. Each number in this system consists of digits which are located at different positions. The position of first digit towards left side of the decimal point is 0. The position of second digit towards left side of the decimal point is 1. Similarly, the position of first digit towards right side of decimal point is -1. The position of second digit towards right side of decimal point is -2 and so on. For example the number 435.6 has 4 * 10 2 + 3 * 101 + 5 * 100 + 6 * 10-1. The positions are often categorized as most significant if on the far left and least significant if on the far right.

On the other hand, Octal Number System is consist of eight digits from 0 to 7. The base of octal system is 8. Each digit position in this system represents a power of 8. The Hexadecimal Number System is consist of 16 digits from 0 to 9 and A to F. The alphabets A to F represent decimal numbers from 10 to 15. The base of this number system is 16.

It will be important to know how conversions can be done from one system to the other and particularly to binary system.

Binary Number System

All data in a computer system consists of binary information. 'Binary' means there are only 2 possible values: 0 and 1. One binary digit (0 or 1) is referred to as a bit, which is short for binary digit. These bits can be grouped together into larger chunks to represent data. Computer designers use eight bit chunks called bytes as the basic unit of data. Computer manufacturers express the capacity of memory and storage in terms of the number of bytes it can hold.
A single byte can represent many different kinds of data. What data it actually represents depends on how the computer uses the byte. Though we use the decimal number system when we input numbers as data or instructions, the computer represents these data as 0 and 1.

Example: For the decimal number 252 we have:

252

1111100

Decimal (Base 10)			Binary (Base 2)							
MS		LS	MS							LS
2	5	2	1	1	1	1	1	1	0	0
100	10	1	128	64	32	16	8	4	2	1
102	101	100	27	26	25	24	23	22	21	20

Adopted and retrieved from Representing signed integers by Dr. Dulay Oct. 2007 available at

https://www.doc.ic.ac.uk/~eedwards/compsys/0_Notes1_Integers.doc

Here MS stands for Most Significant and signifies the leftmost digit or bit. LS stands for Least Significant and signifies the rightmost digit or bit.

The base of a number can be made to be explicit, for example

1410= 168= 11102 =0E16

Decimal to Binary Conversion (Division)

One conversion technique is the repeated division by 2 and taking then recording the remainder.

Example: What is 9810 in binary?

(Adopted and retrieved from Representing signed integers by Dr. Dulay Oct. 2007 available at

https://www.doc.ic.ac.uk/~eedwards/compsys/0_Notes1_Integers.doc)

Steps:

- Divide the number by 2 giving the quotient and the remainder.
- Repeat previous step with the new quotient until a zero quotient is obtained.
- The answer is obtained by reading the remainder column from the bottom to the top.

	Quotient	Remainder
98 ÷ 2	49	0
49 ÷ 2	24	1
24 ÷ 2	12	0
12 ÷ 2	6	0
6 ÷ 2	3	0
3 ÷ 2	1	1
1 ÷ 2	0	1

Answer: 110 00102 reading the remainder column from bottom to the top MS-bit to LS-bit

Octal Number System

(Adopted and retrieved from Representing signed integers by Dr. Dulay Oct. 2007 available at

https://www.doc.ic.ac.uk/~eedwards/compsys/0_Notes1_Integers.doc)

Conversion of binary to octal

The rule of thumb is that you start from the least significant end and for each group of 3 bits represent 1 octal digit (called an octet). For example,

000 (0), 001 (1), 010 (2), 011 (3), 100 (4), 101 (5), 110 (6), 111 (7)

Consider the following examples

Example 1 : What is 101012 in octal?

10	101
2	5

(You group 101 right most and 10 left most and get 25)

Example 2: What is 3578 in binary?

3	5	7
011	101	111

Hexadecimal Number System

Hexadecimal system is used by programmers to represent long binary values. The symbols 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F are used to represent the digits for this. The letters A to F can be in uppercase or lowercase. We start from the least significant end of the binary system and count 4 bits to represent 1 hex digit. The correspondence between the hexadecimal, binary and decimal is shown in the following illustration (Adopted and retrieved from Representing signed integers by Dr. Dulay Oct. 2007 available at https://www.doc.ic.ac.uk/~eedwards/compsys/0_Notes1_Integers.doc)

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binary	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111

Example 1.

Take an example of converting 25210 to hexadecimal. We first convert 252 base 10 to base 2 and get 1111 11002 . We then group them into 4 bits from the left and have (1111)=F and (1100) =C. 25210 = FC 16 .(Adopted and retrieved from Representing signed integers by Dr. Dulay Oct. 2007 available at

https://www.doc.ic.ac.uk/~eedwards/compsys/0_Notes1_Integers.doc)

You can convert binary to hexadecimal number by grouping 4 bits binary numbers

Example 1

0(1101)(1101)2 = 0110 1110 group of 4 bits

= 6 E16

11010011012 = 0011 0100 1101

= 3 4 D16

Example 2

1011100.10001012 = 0101 1100.1000 1010

= 5 12 . 8 1016

= 5C.8A16

Conversion of hexadecimal number to octal number

Example

3DE16 = 00111110111102

Group 3 bits together from left to right then change each grouping to decimal number

= 001 111 011 110

= 1 7 3 68

Addition of Binary Numbers

In the binary system the rules of binary when adding the number are as follows:

0 +0 = 0

0 + 1 = 1

1 + 1 = 1

1+ 1 = 0 with a carry of 1

Example

a) Add $1001_2 + 0101_2$

Worked Solution

1001	9
0101	5
1110_2	14_{10}

b) Add $0111_2 + 0101_2$

Worked Solution

0111	7
0101	5
$1\ 100_2$	12_{10}

Binary Subtraction

Binary Subtraction

Example

a) 1110 – 0101

Worked solution

1110	14
0101	5
1001	9

b) 0101 – 0111

0101	5
0111	7
1110	-2

which is not true since the smaller value is subtracting bigger value computer use 2's complement to handle negative the solution as shown below.

Use 2's complement method

Complement means superimposing the largest value.

0101	0101
0111 complement	1000
	1101 complement this 0010
	0010 which is true

Advantages of the two's complement format

1.Handles subtraction of integers as addition.

2.Therefore computer does not requires different circuits to handle addition and subtraction

Learning Activity 1

Details of the activity: Number system conversion [Estimated Time 2 hour]

Before attempting this activity, you read

1. http://www.tutorialspoint.com/computer_fundamentals/computer_number_system.htm - number system

2. http://www3.ntu.edu.sg/home/ehchua/programming/java/datarepresentation.html - A Tutorial on Data Representation: Integers, Floating-point Numbers, and Characters

3. http://www.edupub.gov.lk/Administrator/English/10/ICT%20g-10%20E%20new%20syllbus/Chapter%203.pdf- Data Represenation methods in a computer system, chapter 3

Try the following conversions and see if the answers are as given.

1.Direct conversions

a. Convert binary number 10102 to decimal number Answer 10_{10}

b. Convert binary number 110012 to decimal number Answer 25_{10}

c. Convert binary number 11111112 to decimal number Answer 127_{10}

d. Convert 111 010 1012 into base ten Answer 25_{10}

2. Construct a table of numbers from 1 through 16 in decimal numbers and convert them into octal, binary and hexadecimal equivalent.

Adopted and retrieved from Chapter 6: Exercises and answers, Retrieved March 14, 2016, from web.nuu.edu.tw/~carlu/ecp.../Ans_CH06.doc and can also be seen http://www.rapidtables.com/math/number/Numeral_system.htm Rapid tables reference and tools. (n.d.))

Answer:- Expected table would look like

binary	*octal*	*decimal*	*hexadecimal*
000	0	0	0
001	1	1	1
010	2	2	2
011	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	20

Retrieved March 14, 2016, from web.nuu.edu.tw/~carlu/ecp.../Ans_CH06.doc and can also be seen http://www.rapidtables.com/math/number/Numeral_system.htm Rapid tables reference and tools.)

Assessment

The Number system

1. Evaluate the the following

   a. Convert 73_{10} to base 2 Answer 1001001_2

   b. Convert 111_{10} to base 2 Answer 1101111_2

   c. Convert 221_{10} into base 2 Answer 11011101_2

2. Evaluate the following

   a. Convert $6B9_{16}$ to base 2 Answer 011010111001_2

   b. Covert $6D.3A_{16}$ to base 2 Answer $0110\ 1101.\ 00111010_2$

   c. Convert $5B.3A_{16}$ to base 8 Answer $1\ 3\ 3\ .\ 1\ 6\ 4_8$

3. Compute the following

   a. $1010_2 + 1101_2$ Answer 10111_2

   b. $1110_2 - 0101_2$ Answer 1001_2

Activity 2

Introduction

Logic Gates

In computer science, the Boolean data type is a data type, having two values (usually denoted true and false), intended to represent the truth values of logic and Boolean algebra. It is named after George Boole, who first defined an algebraic system of logic in the mid 19th century. When one wants to build a computer, he/she begins with digital logic design. Nearly every computer is built using digital logic.

Boolean algebra is the mathematics that is used to analyze digital gates and circuits. The variables used have only two possible values, a logic "0" and a logic "1". A boolean expression, on the other hand, may have an infinite number of variables each labelled to represent input to the expression. A variable of the primitive data type boolean can have two values: true and false (Boolean literals).

For example, we may have the variables A,B,C which may be used to form a logical expression A + B = C. Each of these variables can individually have the values of either 0 or 1.

In computer operation with binary values, Boolean logic can be used to describe electromagnetically charged memory locations or circuit states that are either charged (1 or true) or not charged (0 or false).

Boolean functions may be practically implemented by using electronic gates. The following points are important to understand:

- Electronic gates require a power supply.
- Gate INPUTS are driven by voltages having two nominal values, e.g. 0V and 5V representing logic 0 and logic 1 respectively.
- The OUTPUT of a gate provides two nominal values of voltage only, e.g. 0V and 5V representing logic 0 and logic 1 respectively. In general, there is only one output from a logic gate except in some special cases.

Logic gates are the basic building blocks of any digital system and must be designed. It is an electronic circuit having one or more input and only one output. The relationship between the input and the output is based on a certain logic. Based on this, logic gates are named as

1. AND gate,
2. OR gate,
3. NOT gate,
4. NAND gate,
5. NOR gate,
6. EXOR gate etc.

For further reading: on logic gate available at

1. http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/
2. http://www.slideshare.net/pong_sk1/basic-gates-and-functions

Example:

Left hand side represents boolean function and right hand side represent boolean expression

The left side represents the output, say, Y and it can then be stated that

In this section of the unit, you will learn about the characteristics of the various gates, their design and how they are.

Truth Table

A truth table is a table which is used to show the truth values of a sentence, known as compound sentence, that is formed by using the logical connectives:-

- AND - both the variables are true
- NOT- the variables is true
- OR- either of them is true.

This is true for every possible set of combinations of the truth values of the sentences that form the compound sentence. The main purpose of a truth table is to show the function of a logic gate. These values, associated with the function, are normally given the logic values of 0, for false and 1 for true.

Thus, the following table shows the rules that apply:-

A	-A	B	-B
0	1	1	0
1	0	0	1

It is also true that A + B = 1 except when A =0 and B =0. This one says that either A is true (1) or B is true (1) or both are true. Whenever the two of them are false then A + B ≠ 1;

A * B =0, that is both A and B are false and the only exception is when A=1 and B =1 (both are true);

A * B =1, that is both are true.

The behavior of any logic gates can be understood by examining the truth tables. You can find out how the inputs into the logic gate will be related to the output by using the rules that are used to establish the truth tables. Here, the inputs are shown in the left column of the table and this will include all the various input combinations. To make this easier, the inputs are normally mapped onto the binary logic values of 1 and 0. The gate output(s) are normally shown on the right hand side of the column.

Truth tables can be constructed by following the steps listed below:-

1.List the variables in ascending order of alphabet.

2.Create the rows where the number of rows will be 2 n where n represent number of variables

3.Create the column headings to be the same as the names of the variables.

4.Beginning from the rightmost column, insert alternate values of 0's (False's) and 1's (True's) in the rows until you exhaust the rows.

5.You move to the immediate left column and insert pair of alternate 0's (False's) and 1's (True's) in the rows until you exhaust the rows.

6.Continue to the remaining columns while doubling the 0's and 1's until all the columns have been completed.

At the end of the activity, you will notice that the first horizontal line (row) will have all 0's and the last will have all 1's. You will also notice that leftmost column will be divided evenly where the first half will have all 1's and the second all 0's.

For example, suppose we have only one variable, say, p. This has two possible truth values, 1 and 0. We can then construct a truth table like the one shown in Table 2.1.

p
1
0

Table 2.1: Truth table of one variable

If we have two possible variables, say, p and q. there are four possible truth value combinations, that is, 11, 10, 01 and 00. The net effect is that there will be four rows 22 as shown in Table 2.2.

p	q
1	1
1	0
0	1
0	0

Table 2.2 : Truth table for two variables

AND Gate

This gate got its name because of the logical "AND" operator that requires that the values of the participating variables be either both true or both false. The output will be if both inputs are TRUE else it will be "false".

In the Logic Diagram below, shows the AND gate.

Adapted from Basic gates and functions. Retrieved September 03, 2012, from http://www.slideshare.net/pong_sk1/basic-gates-and-functions and Logic gates. Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

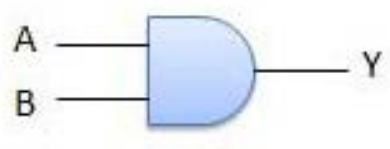


Figure 2.1- AND Gate Logic diagram

Truth Table

INPUT		OUTPUT
A	B	AB =Y
0	0	0
0	1	0
1	0	0
1	1	1

Table 2.3- Truth Table for AND Gate

OR Gate

Adapted from Basic gates and functions. Retrieved September 03, 2012, from http://www.slideshare.net/pong_sk1/basic-gates-and-functions and Logic gate. Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

The OR gate gets its name from the use of the "OR" logical operator. Here, the output will be 1 (TRUE) if either or both the inputs are "TRUE". If both inputs are "false," then the output is "false."

Logic diagram

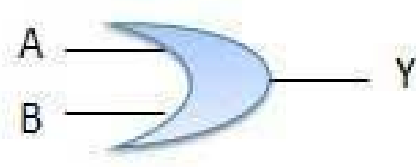


Figure 2 .2- OR Gate

Truth Table

Inputs		outputs
A	B	A+B=Y
0	0	0
0	1	1
1	0	1
1	1	1

Table 2.4- Truth table for OR Gate

NOT Gate

Adapted from Basic gates and functions. Retrieved September 03, 2012, from http://www.slideshare.net/pong_sk1/basic-gates-and-functions and Logic gates. Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

A logical inverter , sometimes called a NOT gate to differentiate it from other types of electronic inverter devices, has only one input. It reverses the logic state.

Logic diagram

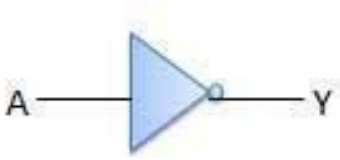


Figure 2.3- NOT Gate

Truth Table

INPUT	OUTPUT
A	Y (NOT A)
0	1
1	0

Table 2.5: NOT Gate truth table

NOR Gate

Adapted from P. S. (n.d.). Basic gates and functions. Retrieved September 03, 2012, from http://www.slideshare.net/pong_sk1/basic-gates-and-functions and Logic gates. Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

The NOR gate is a combination OR gate followed by an inverter. Its output is "true" if both inputs are "false." Otherwise, the output is "false."

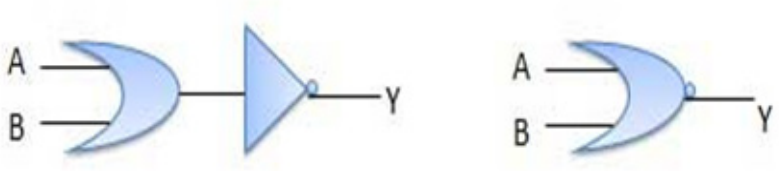


Figure 2. 4: NOR Gate

Inputs		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Truth Table

Logic gates Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

XOR Gate

Adapted from P. S. (n.d.). Basic gates and functions. Retrieved September 03, 2012, from http://www.slideshare.net/pong_sk1/basic-gates-and-functions and Logic gates. (n.d.). Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

The XOR (exclusive-OR) gate acts in the same way as the logical "either/or." The output is "true" if either, but not both, of the inputs are "true." The output is "false" if both inputs are "false" or if both inputs are "true." Another way of looking at this circuit is to observe that the output is 1 if the inputs are different, but 0 if the inputs are the same

Logic diagram

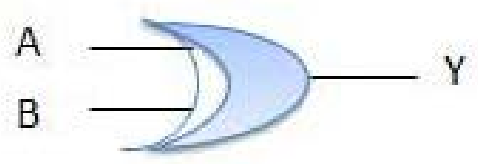


Figure: 2.5: XOR Gate

Truth Table

Inputs		Output
A	B	A ⊕ B
0	0	1
0	1	0
1	0	0
1	1	1

Table: 2.7 :XOR Gate truth table

Adapted from Basic gates and functions. Retrieved September 03, 2012, from http://www.slideshare.net/pong_sk1/basic-gates-and-functions and Logic gates Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

NAND Gate

This kind of gate works like the AND gate only that the AND gate must be followed by a NOT gate. The principle for this one is that the output is false if both the inputs are "TRUE", otherwise the output is "FALSE". Adapted from Basic gates and functions. Retrieved September 03, 2012, from http://www.slideshare.net/pong_sk1/basic-gates-and-functions anf Logic gates.Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

Figure:2.6 NAND Gate

Truth table

Inputs		Output
A	B	NOT(A and B)
0	0	1
0	1	1
1	0	1
1	1	0

Table 2.8 NAND Gate table

Using combinations of logic gates, complex operations can be performed. In theory, there is no limit to the number of gates that can be arrayed together in a single device

- Availble on: http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/TruthFrameSet.htm
- http://site.iugaza.edu.ps/amarasa/files/Lab-11.pdf
- http://studystuff9.blogspot.com/ and Logic gates. Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

Lab Work and Learning Activity 2

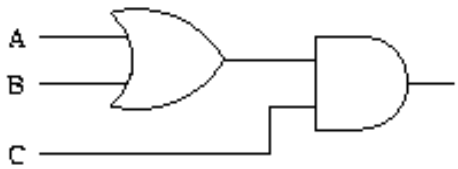
Learning activity details:Design and Construction of Logic gates- [Estimated Time 13 hours).

In this activity, you are to construct and give a logic circuit diagram for each of the following;You may wish to start by constructing truth tables for each first.

1. Supposing you are given the input below, what would be its logic gate?

(A + B)C

Solution

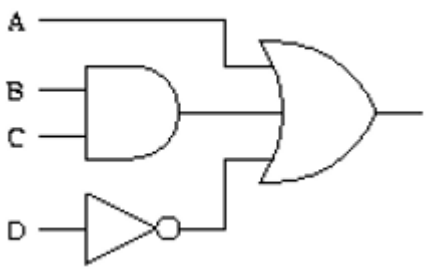


Logic gates. (n.d.). Retrieved April 28, 2016, from http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm

2.Construct a logic circuit for the input below

Solution

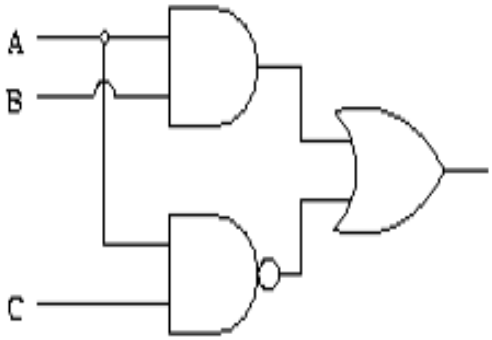
$A + BC + \overline{D}$



3. How would a logic circuit for the input below look like?

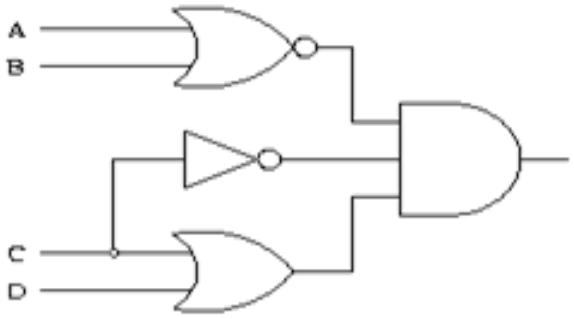
Solution

$AB + \overline{AC}$



4. Construct a logic circuit for the input below

Solution



Learning Activity 3

Learning activities details : Construction of truth tables [Estimated 3 hours]

Consider the following statements. (adapted from Logic. (n.d.). Retrieved April 28, 2016, from http://college.cengage.com/mathematics/berresford/calculus_finite/2e/shared/logic/chapter-l.pdf)

Let X represent the statement "Peter ordered codfish," and let Y represent the statement "Peter ordered ice cream." Take it that ~Y means John did not order ice cream; X ∧ Y John ordered codfish and ice cream (both); V represents either of them. Represent each sentence symbolically using X, Y, and logical connectives the following. Draw an equivalent truth table.

a. Peter did not order ice cream.

b. Peter ordered codfish or ice cream.

c. Peter ordered codfish but not ice cream.

d. If Peter did not order codfish, then he ordered ice cream.

Solution

a. ~Y

b. X V Y

c. X Λ ~ Y

d. ~ X → Y

Truth Table

q	~q
1	0
0	1

2. Let X represent the statement "the yard is large," and let Y represent "the house is small." Express each symbolic statement as a sentence. Draw and equivalent truth table.

i. ~ X Answer: the yard is not small

ii. ~X Λ Y Answer:The yard is not small same as the the house

iii. X Λ Y Answer: the yard is small and the house is small.

(adapted from Logic. (n.d.). Retrieved April 28, 2016, from http://college.cengage.com/mathematics/berresford/calculus_finite/2e/shared/logic/chapter-l.pdf)

Unit Assessment

1. Use 2's Complete to evaluate the following

   a. $0111_2 - 1000_2$

   b. Subtract 4 from 12

2. The output of an AND gate with three inputs, A, B, and C, is HIGH .What are the inputs?

3. If a 3-input NOR gate has eight input possibilities, how many of those possibilities will result in a HIGH output?

4. If a signal passing through a gate is inhibited by sending a LOW into one of the inputs, and the output is HIGH. What is the logic gate?

5. The logic gate that will have HIGH or "1" at its output when any one of its inputs is HIGH . What is the logic gate?

Grading Scheme

This assessment is worth 5 % of your total course mark as follows

Each question carries 1 %

Feedback

You should check if you got the following answers.

1. Use 2's Complete to evaluate the following

   a. $01112 - 1000_2$ Answer 0001_2

   b.Subtract 4 from 12 Answer 1000_2

2. The output of an AND gate with three inputs, A, B, and C, is HIGH .What are the inputs?
Answers: A = 1, B = 1, C = 1

3. If a 3-input NOR gate has eight input possibilities, how many of those possibilities will result in a HIGH output?

Answer.1

4. f a signal passing through a gate is inhibited by sending a LOW into one of the inputs, and the output is HIGH. What is the logic gate?

Answer.NAND Gate

5. The logic gate that will have HIGH or "1" at its output when any one of its inputs is HIGH . What is the logic gate?

Answer OR gate

Unit Summary

The unit has exposed you to the basics of numbering systems, use of decimal, octal, binary and hexadecimal number system. The computer understands only binary number and that all other number systems are to be converted into the binary number systems. The techniques of conversion have been covered.

The boolean algebra is the basis of logic circuits. The truth tables are constructed based on the values of the input variables in the boolean expressions. You can now identify, describe and analyze the gate symbols for the Boolean operations AND, OR, NOT and XOR.

Unit Readings and Other Resources

The readings in this unit are to be found at course level readings and other resources.

1. http://www.edupub.gov.lk/Administrator/English/10/ICT%20g-10%20E%20new%20syllbus/Chapter%203.pdf Data Representation Methods in the Computer system

2. http://www.rapidtables.com/math/number/Numeral_system.htm Rapid tables reference and tools. (n.d.).

3.http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/index.html #notgate: Basic Gates and Functions

4. http://www3.ntu.edu.sg/home/ehchua/programming/java/datarepresentation.html - A Tutorial on Data Representation: Integers, Floating-point Numbers, and Characters

5. http://www.ee.surrey.ac.uk/Projects/CAL/digital-logic/gatesfunc/index.html#notgate : Basic Gates and Functions

6. http://www.tutorialspoint.com/computer_logical_organization/logic_gates.htm- logic gates

7. http://college.cengage.com/mathematics/berresford/calculus_finite/2e/shared/logic/chapter-l.pdf - logic

Unit 3: Micro Architecture

Introduction

In this unit, you will learn about the organization of the computer, the part of computer architecture that defines the data paths, data processing and storage elements, as well as how these elements should be implemented in the ISA discussed in the other units. Computer instructions are composed of an operation code, also known as opcode, and one or more operands. An operand is something which is acted on by the operation. It can be a constant value, a register identification code, or the address of a location in memory or of an I/O device.

You already know that computers only understands binary of 1's and 0's. The simplest instruction, opcode, is known as the machine code and is executed directly by the CPU. This machine code allows the computer to perform very basic but essential tasks. The results of these instructions are stored in a register known as accumulator.

Unit Objectives

Upon completion of this unit you should be able to:

1. Explain the structure of the interconnection structures of the computer.
2. Explain how control unit interprets a machine-level instruction – either directly or as a microprogram.
3. Identify internal operations of a computer

Key Terms

An address bus is a computer bus (a series of lines connecting two or more devices) that is used to specify a physical address.

Data bus is an internal pathway across which data are transferred to and from the processor or to and from memory (http://www.pcmag.com/encyclopedia/term/40745/data-bus)

A program counter is a register in a computer processor that contains the address (location) of the instruction being executed at the current time. As each instruction gets fetched, the program counter increases its stored value by 1. After each instruction is fetched, the program counter points to the next instruction in the sequence. When the computer restarts or is reset, the program counter normally reverts to 0.(http://whatis.techtarget.com/definition/program-counter)

In a computer, **the Memory Address Register (MAR)** is a CPU register that either stores the memory address from which data will be fetched to the CPU or the address to which data will be sent and stored. In other words, MAR holds the memory location of data that needs to be accessed. (https://en.wikipedia.org/wiki/Memory_address_register)

The Memory Data Register (MDR) or Memory Buffer Register (MBR) is the register of a computer's control unit that contains the data to be stored in the computer storage (e.g. RAM), or the data after a fetch from the computer storage. (https://en.wikipedia.org/wiki/Memory_data_register))

In computing, **an instruction register (IR)** is the part of a CPU's control unit that stores the instruction currently being executed or decoded. (https://en.wikipedia.org/wiki/Instruction_register)

Introduction

As explained in Unit 1, functional components of a computer system include input unit, central processing unit and the output unit. These units are connected through the motherboard as explained in unit 1.There are a variety of interconnection structures that have been designed and in use, but the most common is the bus and various multiple-bus structures. An important feature of the bus is that it is a shared transmission medium where more than one device can be connected to it. The moment a device is connected a signal is then transmitted and is received by all the devices connected to the bus.

The transmission is such that only one device at a time can do the transmission. This is to avoid confusion which may result if more than one device is allowed to transmit. In modern computers, a bus is consist of multiple pathways, known as lines, that are used to transmit signals. A system bus may a times have about 50 to 100 separate lines where each line is dedicated to a specified function or has a particular meaning. For example, there are three functional groups which are data, address and control lines.

The diagram below shows such a structure and the functional groups.

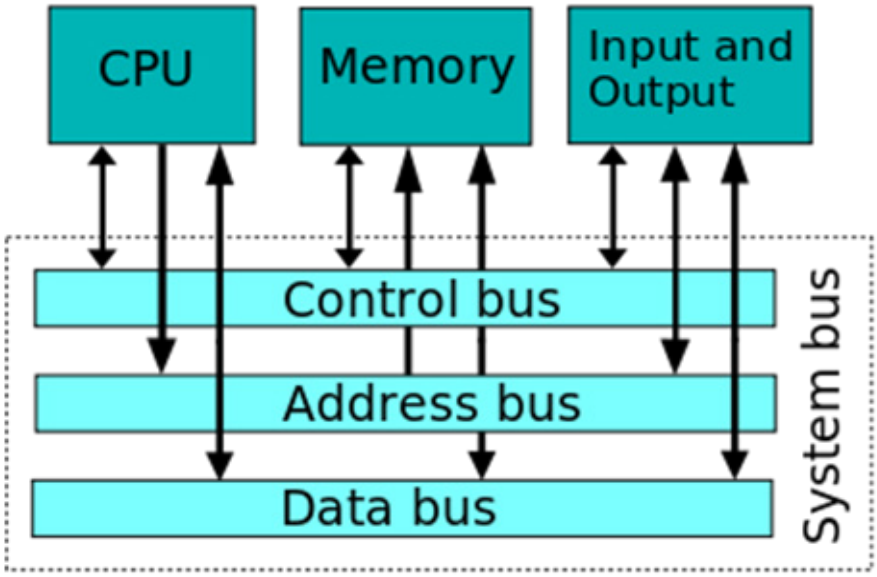


Figure 3.1-computer Bus:

Available at: http://en.wikipedia.org/wiki/System_bus.The data bus, also known data lines, may have 32, 64, 128 or even more such separate lines. The number of lines represent the width of the data bus. Here, a data line can only carry 1 bit at a time and therefore, the number of lines is an indicator of how many bits can be transferred at a time. The other group of lines is the address bus (lines) and are used to show the source or the destination on the data bus. The width of the address bus is used to determine the memory capacity of the system.

The other group is the Clock control lines which are used to synchronize operations and initialize all modules.

In case the CPU or the memory or the I/O wants to transmit data, there are two important things that must occur. The use of the bus must be requested and obtained and that the data must transferred via the bus address lines.

Learning Activities

Before doing this activity, you are to read the following materials as listed below [10 hours]

1. http://www.tutorialspoint.com/computer_fundamentals/index.htm- Computer Fundamentals- (Read from computer types to computer hardware)

2. http://simple.wikipedia.org/wiki/Computer_architecture - computer architecture

3.http://en.wikiversity.org/wiki/Computer_architecture_and_organization - Computer architecture and organization

You will also spend some time, about seven hours, to look for related materials in order to competently complete the activities given here in this module.

Activity 1

Activity Details : Interconnection structures - The Bus [5 hours]

In this activity, you are to open at least two computers of different computer architecture models complete the following:-

1. Name the computer architecture widely used by modern computers

2. From the computer architecture model you are using identify the following:

a) Number of data buses

b) Number of address buses

c Number of control buses

3. Identify the type of connection for each I/O device and list the type of connection, e.g ISA, PCI, USB, COM port, etc.

4.Indicate what kind of data each of the I/O devices may send through the data bus.

Conclusion

There are different buses, input/output devices, ports, USB, COM ports and that these are different for different computer architecture models and that some are standard like ports, USB, COM ports.

Instruction cycle (The Fetch-Execute Cycle)

Introduction

This is the process where a computer retrieves program instructions from its memory, examines them and determines what actions the instructions requires and eventually executes (performs) those actions. This activity is repeated from the time the computer is switched on to when it is switched off. This process is sequential in that each instruction is processed to completion before another one is started. In modern machines, there are concurrent executions where instructions are processed in parallel.

Central Processing Units (CPUs) have different instruction cycles because of different instruction sets. Instruction Cycle, " Instruction Cycle. Web. 15 Mar. 2016. Available at http://www.digplanet.com/wiki/Instruction_cycle

The implementation of what has been described here in the fetch and execute cycle will be discussed in unit 4, Instruction Set Architecture (ISA).

Essential Registers for Instruction Execution

There are four registers that are essential to instruction execution and they are:- (Adopted from Introduction to the CPU, Retrieved May 3, 2016, from http://nptel.ac.in/courses/Webcourse-contents/IIT- and W. S. (n.d.). Computer functions and interconnections. Retrieved May 4, 2016, from http://www.guswnsla1223.tistory.com/attachment/cfile29.uf@153C97234AE9564A37B69E.pdf

copyright @2005)

- Program Counter (PC)- This register contains the address of an instruction that is to be fetched. The program counter is typically incremented (made to point to the next instruction) once a given instruction has been fetched. The PC, therefore, will always point at the next instruction to be fetched.
- The Instruction register (IR)- This is the register that contains the most recently fetched instruction. The opcode (operation code) and operand specifies are analyzed once the fetched instruction is loaded onto the IR.
- The memory address register (MAR)-this register holds the address of a location of the main memory from where the information is to be fetched or stored. The contents of this memory is directly connected to the address bus.
- The memory buffer register (MBR)-This register contains a word of data to be written to the main memory or that it contains the word that was most recently read. The contents of this register are directly connected to the data bus.

Apart from the above dedicated registers, there are temporary registers that are not visible to the user and only serve as input/output registers for the arithmetic logic unit (ALU). They exchange data with MBR and user visible registers.

Program Execution

The instructions that form a program to be executed are loaded onto the main memory in sequential locations (Adopted from Introduction to the CPU, Retrieved May 3, 2016, from http://nptel.ac.in/courses/Webcourse-contents/IIT-). To execute this program, the central processing unit is then to fetch the instructions one after the other. It will then decode the instructions and thereafter performs the actions required. The detailed actions are described below for each of the stages of the fetch-decode and execute sequence.

The instruction must be fetched and loaded to the instruction register. The instruction is expected to contain bits that details the actions that are to be performed by the processor. The processor is then expected to interpret the instruction to know the actions that are to be done. In general, the actions that can be performed can be grouped into four categories(Adopted from Introduction to the CPU, Retrieved May 3, 2016, from http://nptel.ac.in/courses/Webcourse-contents/IIT-). They are:-

- Processor-memory: This is where data is transferred from processor to memory or from the memory to the processor.
- Processor-I/O: This is where data may be transferred to/from a peripheral device and is to be done by transferring between the processor and the module that deals with the input/output.
- Data processing: Arithmetic or logic operations may be performed on the data.
- Control: There could be an instruction that may make a specification that the sequence of the executions of the actions may be changed.

You will notice that the main task will consist of changing from fetching the instruction and the execution of the actions associated with the instruction. The instruction must be examined to establish if there are any indirect addressing that may be involved. If there is one, then the operands that may be involved must be fetched using the indirect addressing mode. It should also be mentioned that the performance of actions associated with a given instruction (execution of the instruction) may require more than one reference to memory. In some cases, an instruction may include a specification of an input/output operation.

The fetch and execute can be seen as a state of activities that is composed of the steps explained below, some states may be null while others may be done more than once (Adopted from Introduction to the CPU, Retrieved May 3, 2016, from http://nptel.ac.in/courses/Webcourse-contents/IIT- and also in http://cs323.wikispaces.com/file/view/Ch3.pptx).

- Instruction Address Calculation (IAC): This is a state where the address of the next instruction is calculated and this normally involves changing the program counter by adding a fixed number of the address of the previous instruction.
- Instruction Fetch (IF): This is the state where the instruction is read from the memory location into the processor.
- Instruction Operation Decoding (IOD): This is the state where analysis is done to establish the type of operation to be performed and the operands that are involved.
- Operand Address Calculation (OAC): This one here determines the address of the memory if the operation is to use an operand.
- Operand Fetch (OF): This one is for fetching the operand from memory and have it read from the input/output.
- Data Operation (DO): This is the state where the operation is performed as indicated in the instruction.
- Operand Store (OS): This is the state where the result of the action is written into the memory or taken out to the I/O. (Computer functions and interconnections.
- Retrieved May 4, 2016, from http://www.guswnsla1223.tistory.com/attachment/cfile29.uf@153C97234AE9564A37B69E.pdf copyright @2005)

The implementation of the sequence is also given in unit 4 and you are encouraged to understand the actions that form the steps of the sequence.

1.Detailed actions in the Fetch the Instruction

The first step of the Instruction Cycle is called the Fetch instruction Cycle. The Control Unit fetches the instruction's address from the memory unit using following the steps: (Adapted from Instruction Cycle." Instruction Cycle. Web. 15 Mar. 2016. Available at http://www.digplanet.com/wiki/Instruction_cycle and http://web.cse.ohio-state.edu/~reeves/CSE2421au12/SlidesDay35.pdf)

a. The CPU sends PC to the MAR and sends a READ command on the control bus

b. In response to the read command (with address equal to PC), the memory returns the data stored at the memory location indicated by PC on the data bus.

c. The CPU copies the data from the data bus into its MDR

d. A fraction of a second later, the CPU copies the data from the MDR to the Instruction Register (IR)

e. The PC is incremented so that it points to the next instruction in memory. This step prepares the CPU for the next cycle.

2.Detailed actions in the Fetch phase

(ComputerArchitecture: Fetch and execute cycle adopted and available at http://www.inf.uni-konstanz.de/dbis/teaching/ws0304/computing-systems/download/rs-02.pdf and (Adopted from Introduction to the CPU, Retrieved May 3, 2016, from http://nptel.ac.in/courses/Webcourse-contents/IIT-))

a. The address in the CPU register IP is transmitted via the address bus to the memory unit's MAR: (IP points at individual bytes [= 8 bits] in memory):

   IP→MAR

b. IP is incremented to point at the next program instruction, ready for the next cycle

   IP ++

c. Memory selects addressed location and copies its contents onto the data bus; CPU loads received data into IR:

   (MAR) → IR

d. CPU starts decoding the instruction in IR

(Available at http://web.cse.ohio-state.edu/~reeves/CSE2421au12/SlidesDay35.pdf)

3.Detailed actions in the Decode the Instruction

The decoding process allows the CPU to determine what instruction is to be performed, so that the CPU can tell how many operands it needs to fetch in order to perform the instruction. The opcode fetched from the memory is decoded for the next steps and moved to the appropriate registers. The decoding is done by the CPU's Control Unit. (Adapted from Instruction Cycle." Instruction Cycle. Web. 15 Mar. 2016. Available at http://www.digplanet.com/wiki/Instruction_cycle and http://web.cse.ohio-state.edu/~reeves/CSE2421au12/SlidesDay35.pdf)

4.Detailed actions in the Read the effective address

Read instruction cycle is in deciding which operation it is. If this is a Memory operation - in this step the computer checks if it's a direct or indirect memory operation:

- Direct memory instruction - Nothing is being done.
- Indirect memory instruction - The effective address is being read from the memory.

If this is a I/O or Register instruction - the computer checks its kind and executes the instruction.

(adapted from Instruction Cycle." Instruction Cycle. Web. 15 Mar. 2016. Available at http://www.digplanet.com/wiki/Instruction_cycle , http://web.cse.ohio-state.edu/~reeves/CSE2421au12/SlidesDay35.pdf and ((Adopted from Introduction to the CPU, Retrieved May 3, 2016, from http://nptel.ac.in/courses/Webcourse-contents/IIT-).)

5.Detailed actions in the Execute the Instruction

Execute cycle is where the action of the instruction is performed. If the instruction involves arithmetic or logic, the Arithmetic Logic Unit is utilized. This is the only stage of the instruction cycle that is useful from the perspective of the end user. Everything else is overhead required to make the execute stage happen.

(adapted from Instruction Cycle." Instruction Cycle. Web. 15 Mar. 2016. Available at http://www.digplanet.com/wiki/Instruction_cycle , http://web.cse.ohio-state.edu/~reeves/CSE2421au12/SlidesDay35.pdf and ((Adopted from Introduction to the CPU, Retrieved May 3, 2016, from http://nptel.ac.in/courses/Webcourse-contents/IIT-)

6.Detailed actions in the Store the Result in Memory

The last final instruction cycle is where the result is stored the result in a register or in memory, if there is one. The program is then updated to hold the next instruction location, which is either the next memory location or the address specified by a branch instruction. (adapted from Instruction Cycle." Instruction Cycle. Web. 15 Mar. 2016. Available at http://www.digplanet.com/wiki/Instruction_cycle and http://web.cse.ohio-state.edu/~reeves/CSE2421au12/SlidesDay35.pdf and (Adopted from Introduction to the CPU. Retrieved May 3, 2016, from http://nptel.ac.in/courses/Webcourse-contents/IIT-)

Activity 2

Activity details :- Reflection Activity Instruction cycle(The Fetch-Execute Cycle) [5 hours] In this activity, you are to describe the functions of the following in the fetch-execute cycle:

a. Program counter

b. Memory address register (MAR)

c. Memory data register (MDR)

d. Instruction register (IR)

e. Arithmetic Logic Unit (ALU)

f. Control Unit (CU)

You may wish to refer to your notes and /or materials you found in unit 1. Prepare brief notes on these.

Activity 3

Activity details: Instruction cycle(The Fetch-Execute Cycle) [3 hour]

In this activity you are to identify and describe the characteristics the following in fetch decode execute cycle, provide 3 examples for each of them from different sources.

a. Direct memory instruction

b. Indirect memory instruction

Conclusion

The fetch, decode, execute and store instructions are the steps that must be followed during data processing and you have learnt how step is implemented.

Unit Assessment

1. Where does the Instruction cycle take place ?
2. Why does data need to be fetched? Isn't it okay where it is?
3. Describe the stages of the fetch-execute cycle, stating the components involved and their functions in the cycle.
4. Explain how the computer keep track of instructions?
5. Where does the computer put the instruction it has just fetched?

Grading Scheme

Each question earns 1 mark = 5%

Feedback

The answers to these questions are found in the notes. Find them, compile and send them to your instructor e-mail address.

Unit Summary

In this unit, you have been able to comprehend the structure of the interconnection structures of the computer. You can now explain how a CPU's control unit interprets a machine-level instruction – either directly or as a microprogram. You can use register transfer language to describe internal operations in a computer system.

Unit Readings and Other Resources

The readings in this unit are to be found at course level readings and other resources.

1. http://www.cise.ufl.edu/~mssz/CompOrg/CDA-lang.html Organization of Computer Systems: ISA, Machine Language, and Number Systems

2. http://en.wikibooks.org/wiki/A-level_Computing/AQA/Computer_Components,_The_Stored_Program_Concept_and_the_Internet/Machine_Level_Architecture/Machine_code_and_processor_instruction_set Machine Level Architecture: Machine code and processor instruction set

3. http://teaching.idallen.com/dat2343/10f/notes/410_MachineLevelInstructions.html-Machine Machine level instructions in the General mode.

4. www.csie.nuk.edu.tw/.../Chapter2_Instr...- Chapter 2 Instructions:Language of the Computer

Unit 4. Instruction Set Architecture (ISA)

Introduction

In this unit, you will learn about Instruction Set Architecture. This is the part of the central processing unit that is visible to the programmer who writes the compiler instructions.

Each type of central processing unit is designed to understand a specific group of instructions called the instruction set. This set has an architecture known as instruction set architecture (ISA). An instruction set architecture (ISA) is the interface between the computer's software and hardware and also can be viewed as the programmer's view of the machine as in it defines the codes that a central processor reads and acts upon. An interface is a shared boundary across which two separate components of a computer system exchange information. The exchange can be between software, computer hardware, peripheral devices, humans and combinations of these. (seeD. D. (2007, November). Computer Systems 113 Architecture 110. Retrieved April 12, 2016, from http://www.doc.ic.ac.uk/~eedwards/compsys/0_Notes2_MemoryCPU.doc and in Instruction set explained in, retrieved April 12, 2016,from http://everything.explained.today/Instruction_set/).

The instruction set architecture is an interface but for microprocessors. It particularly concerns itself with the following issues:-

1. What operations can be performed by the computer? This is the instruction set.
2. How are the instructions specified? This one is known as instruction format.
3. Where is the data to be processed located? This is the data storage.
4. How can the data be accessed? This is the addressing mode

In this unit, you will learn how the above concerns are dealt with at the microprocessor interface.

Unit Objectives

Upon completion of this unit you should be able to:

1.Identity key operations that a computer can perform

2.Describe how Instructions are specified.

3.Explain where data is stored in a computer system

4.Explain different addressing modes for accessing stored data

5.Design and implement machine instructions

Key Terms

An instruction set architecture (ISA) is the interface between the computer's software and hardware and also can be viewed as the programmer's view of the machine. It defines the codes that a central processor reads and acts upon

An interface is a shared boundary across which two separate components of a computer system exchange information. The exchange can be between software, computer hardware, peripheral devices, humans and combinations of these. Some computer hardware devices such as a touchscreen can send and receive data through the interface, while others such as a mouse, microphone or joystick are one way only.

An assembly language is a low-level programming language for a computer, or other programmable device, in which there is a very strong (generally one-to-one) correspondence between the language and the architecture's machine code instruction

A memory address is a unique identifier used by a device or CPU for data tracking. This binary address is defined by an ordered and finite sequence allowing the CPU to track the location of each memory byte. Memory addresses are usually allocated during the boot process.

Addressing modes are an aspect of the instruction set architecture in most central processing unit (CPU) designs. The various addressing modes that are defined in a given instruction set architecture define how machine language instructions in that architecture identify the operand (or operands) of each instruction. An addressing mode specifies how to calculate the effective memory address of an operand by using information held in registers and/or constants contained within a machine instruction or elsewhere.

A load/store architecture only allows memory to be accessed by load and store operations, and all values for an operation need to be loaded from memory and be present in registers. Following the operation, the result needs to be stored back to memory. For instance, in a load/store approach both operands for an ADD operation must be in registers.

A register memory architecture allows operations to be performed on memory, as well as registers. In this approach one of the operands for ADD operation may be in memory, while the other is in a register

A compiler is a program that translates a source program written in some high-level programming language into machine code for some computer architecture (such as the Intel Pentium architecture Machine Language Instructions)

What Instructions can be performed (machine language instructions)

Instructions are words of a language that are understood by a machine. They are a set of binary codes that are used to direct the activities of the Central processing unit (CPU).

There are three different types of instructions which are:-

a. Data movement instructions

- The Load instruction has the source being the memory and destination being the register
- The Store instruction has the source being register and destination being memory

b. Arithmetic and logic (ALU) instructions. These are the

Add, Sub, Shift, etc.

- Branch instructions (control flow instructions)
- These are the unconditional or conditional branches

Each of these instructions must have a format and depends on the machine you are using. An instruction format or instruction code is a group of bits used to perform a particular operation on the data stored in a computer. Processor fetches an instruction from memory and decodes the bits to execute the instruction. Different computers may have their own instruction set. An instruction is normally made up of a combination of an operation code and some way of specifying an operand, most commonly by its location or address in memory.

Some operation codes deal with more than one operand; the locations of these operands may be specified using any of the many addressing schemes. Different machines have different instruction set architectures. These architectures are differentiated from one another by the number of bits allowed per instruction (16, 32, and 64 are the most common).

Instruction Format

The instructions are 16 bit and this so that they can fit into main memory word. The instruction is divided into a number of instruction fields that represents a different information for the CPU. Each instruction has a format that is consist of operation code (OPCODE) and in some cases, operands. The commonly used format is of the form as shown below

Memory Address	Machine Codes	Labels	Mnemonics	Operands	Comments

Adapted from Instruction Format.(Retrieved April 13, 2016, from http://gradestack.com/Microprocessors-and/Assembly-Language/Instruction-Format/19312-3912-38125-study-wtw grade stack

1. **Memory address -**is the where the program or instruction is stored.
2. **Machine codes -**are the hexadecimal representation of operation codes
3. **Labels-** is optional and it provides a symbolic name of branch instructions of the instructions; it is normally used for conditional/unconditional jumping.
4. **Mnemonics** states the operation to be executed.
5. **Operands** -for one byte instruction there is no operand, for two byte instruction there is one operand and for a three byte instruction there are two operands separated by a comma.

Addressing Modes

There are different fundamental addressing modes

1. Direct Addressing mode (also called Absolute Addressing)
2. Indirect addressing mode
3. Index Addressing mode
4. Relative addressing mode
5. Autoincrement addressing Mode
6. Auto-decrement addressing mode

(available in Saleh, S. M. (2013, January 17). Http://www.uobabylon.edu.iq/eprints/pubdoc_1_15253_37.doc. Addressing Modes.

Direct (Absolute) Addressing Mode

In this kind of instruction, the source operand represent the value stored in the location whose address is given, 30h. It will then store in the accumulator A. The instruction is then given as MOV A,30h. You can see that it follows the format " Opcode Effective Address of Operand" described at Saleh, S. M. (2013, January 17). http://www.uobabylon.edu.iq/eprints/pubdoc_1_15253_37.doc. Addressing modes

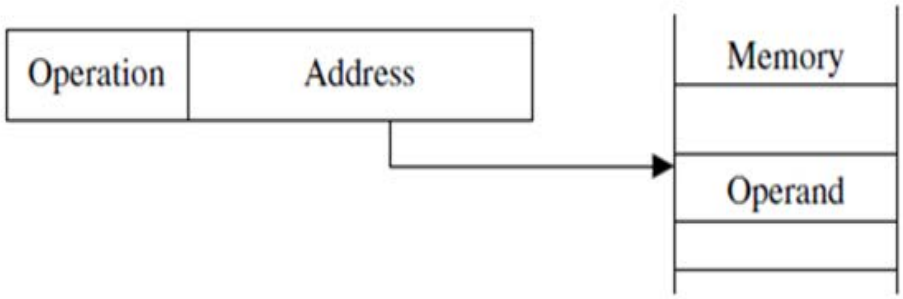


Figure 4.2: Illustration of Direct Addressing Mode.

Adopted and retrieved from instruction set architecture and design available at http://www.slideshare.net/srisumandas/2instruction-set-architecture-design

Indirect Addressing Mode

Indirect mode holds memory location of effective address of the operand.

For example, the instruction LOAD (1000), Ri, this instruction has the memory location 1000 enclosed in parentheses, thus indicating indirection. The meaning of this instruction is to load register Ri with the contents of the memory location whose address is stored at memory address 1000.There are two types of indirect addressing.

1. Register indirect addressing: A register is used to hold the address of the operand,

2. Memory indirect addressing: A memory location is used to hold the address of the operand. The two types are illustrated in figure 4.3

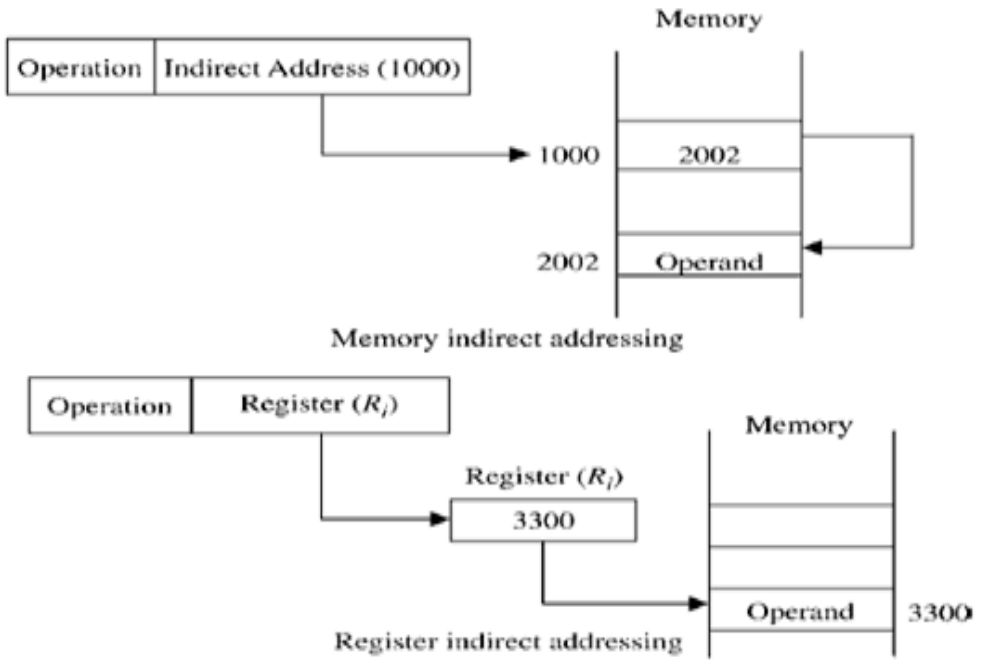


Figure 4.3: Indirect addressing mode Illustration.

Adopted and retrieved from instruction set architecture and design available at http://www.slideshare.net/srisumandas/2instruction-set-architecture-design

Indexed Addressing Mode

In this addressing mode, the address of the operand is obtained by adding a constant to the content of a register, called the index register. Consider, for example, the instruction LOAD X(Rind), Ri. This instruction loads register Ri with the contents of the memory location whose address is the sum of the contents of register (Rind) and the value X. Index addressing is indicated in the instruction by including the name of the index register in parentheses and using the symbol X to indicate the constant to be added. Indexing requires an additional level of complexity over register indirect addressing.

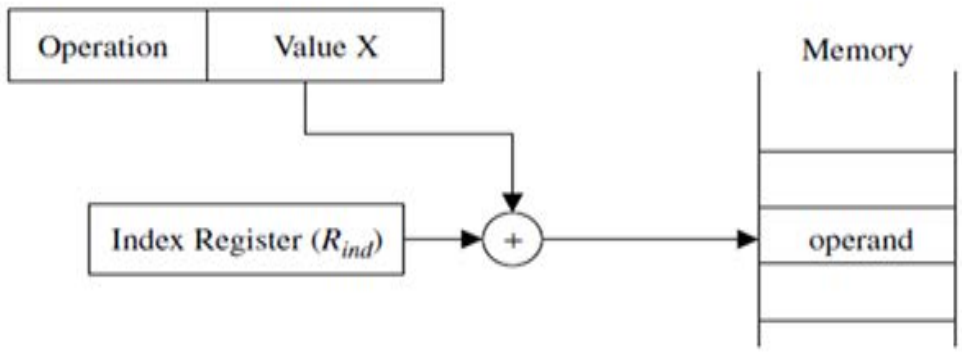


Figure 4.4 Indexed addressing mode Illustration

Adopted and retrieved from instruction set architecture and design available at http://www.slideshare.net/srisumandas/2instruction-set-architecture-design

Relative Addressing Mode

Relative addressing is the same as indexed addressing except that the program counter (PC) replaces the index register.

For example, the instruction LOAD X(PC), Ri loads register Ri with the contents of the memory location whose address is the sum of the contents of the program counter (PC) and the value X. Figure 18 illustrates the relative addressing mode.

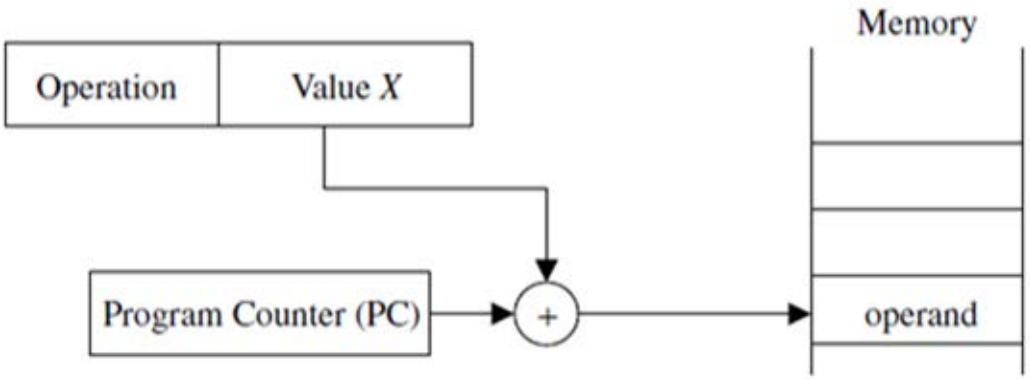


Figure 4.5: Relative Addressing Mode;

Adopted and retrieved from instruction set architecture and design available at http://www.slideshare.net/srisumandas/2instruction-set-architecture-design

Auto Increment Addressing Mode

The content of the autoincrement register is incremented after accessing the operand. The automatic increment of the register's content after accessing the operand is indicated by including a (+) after the parentheses.

For example, the instruction LOAD (Rauto)+, Ri. This instruction loads register Ri with the operand whose address is the content of register Rauto. After loading the operand into register Ri, the content of register Rauto is incremented, pointing for example to the next item in a list of items. Figure 19 illustrates the autoincrement addressing mode.

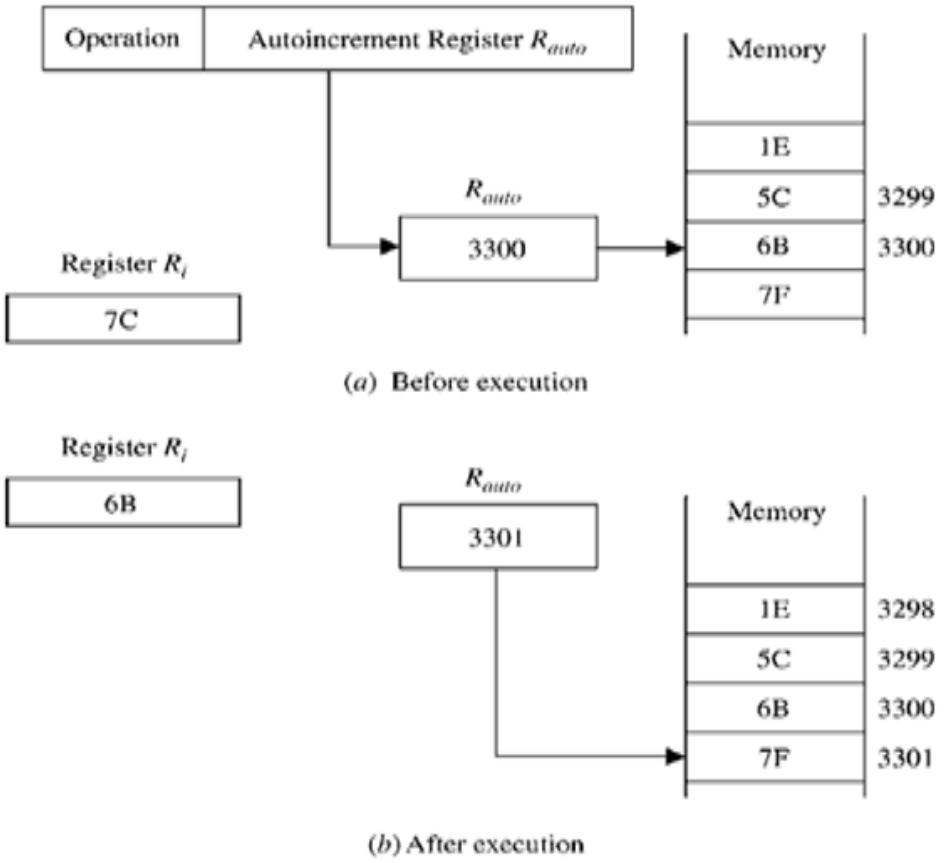


Figure 4.6 : Autoincrement addressing mode;

Adopted and retrieved from instruction set architecture and design available at http://www.slideshare.net/srisumandas/2instruction-set-architecture-design

Auto-decrement Addressing Mode

The auto decrement mode uses a register to hold the address of the operand. The content of the auto decrement register is first decremented and the new content is used as the effective address of the operand. The content of the auto decrement register is decremented before accessing the operand, a (—) is included before the indirection parenthesis .

For example, the instruction LOAD (Rauto), Ri. This instruction decrements the content of the register Rauto and then uses the new content as the effective address of the operand that is to be loaded into register Ri. (Adopted and retrieved from - Instruction set Architecture

available at http://www.slideshare.net/srisumandas/2instruction-set-architecture-design)

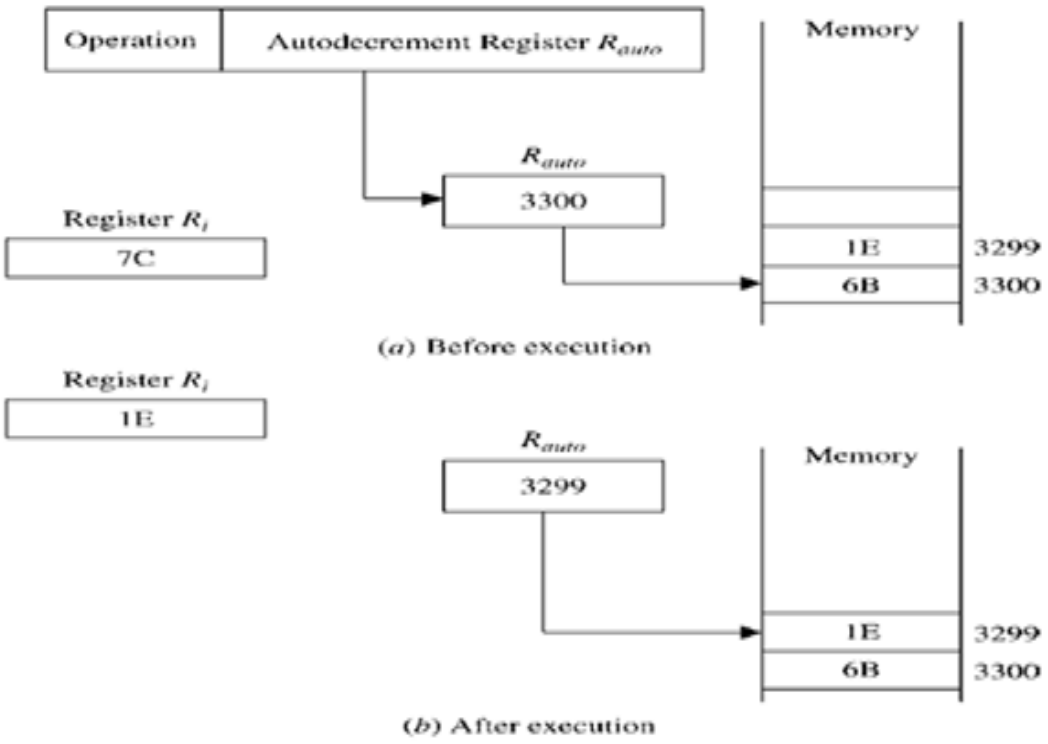


Figure 4.7 Auto Decrement Addressing Mode Illustration;

Learning Activities

Learning Activity 1

Activity Details:- Direct Addressing mode_[1 hour]

Before doing this activity, you are asked to read the following topics in the listed reading materials.

1. JOHN DAINTITH. "instruction format." A Dictionary of Computing. 2004. Encyclopedia.com. 12 Mar. 2016 <http://www.encyclopedia.com>.

2. https://en.wikibooks.org/wiki/A-level_Computing/AQA/Computer_Components,_The_Stored_Program_Concept_and_the_Internet/Machine_Level_Architecture/Machine_code_and_processor_instruction_set -Machine level architecture: Machine and processor instruction set http://www.personal.kent.edu/~aguercio/CS35101Slides/Tanenbaum/CA_Ch05_PartII.pdf- Common Instruction Formats.

3. http://www.kuroski.net/CS221-01/addressing%20modes.doc- addressing modes

In this activity, you will practice how to write direct addressing mode instruction. At the end of each, you will be able to write your own instructions by following the details of each activity.

1. Write machine instructions to compares the two numbers in R1 and R2, and then puts the value

a. 0 in R0 if R1 = R2,

b. 1 if R1 > R2,

c. -1 if R1 < R2.

Finally, store the result to memory location 0x4000.

Learning Activity 2

Activity Details - Indirect Addressing Modes (how is data accessed?) [1 hour]

Before doing this activity, you are asked to read the following topics in the listed reading materials.

1. JOHN DAINTITH. "instruction format." A Dictionary of Computing. 2004. Encyclopedia.com. 12 Mar. 2016 <http://www.encyclopedia.com>.

2. https://en.wikibooks.org/wiki/A-level_Computing/AQA/Computer_Components,_The_Stored_Program_Concept_and_the_Internet/Machine_Level_Architecture/Machine_code_and_processor_instruction_set -Machine level architecture: Machine and processor instruction set

3. http://www.slideshare.net/srisumandas/2instruction-set-architecture-design- Instruction set architecture and design (sections 2.1 to 2.3.2)

4. http://www.personal.kent.edu/~aguercio/CS35101Slides/Tanenbaum/CA_Ch05_PartII.pdf-

5. http://www.kuroski.net/CS221-01/addressing%20modes.doc- addressing modes

Activity Procedure

1. Rewrite the following sequence of code using the register and register indirect addressing modes:

1. LOAD r1, (200)
2. ADD r3, r2, 10(r1)
3. AND r3, r2, @r2

Answer

1. LOAD r1, 200# immediate;
2. LOAD r1, (r1)# register indirect
3. ADD r4, r2, 10# these two instructions replace the
4. ADD r3, r2, r4# ADD in the above sequence
5. LOAD r4, (r2)# and the two simulate the memory
6. AND r3, r2,(r4)# indirect addressing;

2. An integer is stored somewhere in the memory; a pointer to this integer is at address 200. Show how memory indirect addressing is used to increment the number.

Answer

If the machine supports the base address (200) to be in memory we have:

1. LOAD r2, 1
2. ADD r2, r2, @(200)
3. STORE@(200), r2

Learning Activity 3

Reflective Activity details: Addressing Modes [1 hour]

Explain the differences in mode of operation of the following addressing modes

1. Register Addressing

2. Immediate Addressing

3. Direct Memory Addressing

4. Direct offset Addressing

Conclusion

You have learnt that there are various addressing modes that are involved in fetch and execute instructions, what operations to be performed, where to fetch the operands from, where to store the results, and address of the instruction to be fetched next.

Summary

Estimated time [2 hours]

1. Where is the data (operand) if the address mode specifier is:

    a. 000

    b. 001

2. Addressing mode is a way to address an operand. Operand means the data we are operating upon (in most cases source data). It can be a direct address of memory, it can be register names, it can be any numerical data etc. Explain this with a simple data move instruction of 4545 below:

MOV A,30h

3. An array of two integers (each integer = 32 bits) is placed in memory starting with address 100. Show how to increment each element of the array using register indirect addressing mode.

Unit Summary

In this unit you have learnt the concept of an instruction set architecture, ISA, and the nature of a machine-level instruction in terms of its functionality and use of resources (registers and memory).You have been able to characterize the differences between register-to-memory ISAs and load/store ISAs.

You are now able to analyze the relationship between instruction set architecture, micro architecture, and system architecture and their roles in the development of the computer.

Grading scheme

Each question earns 2 % and Total = 6 % for this unit.

Feedback

1. Where is the data (operand) if the address mode specifier is:

    a. 000

    b. 001

Answer

a. In the instruction specifier

b.In the place named in the instruction specifier

2. Addressing mode is a way to address an operand. Operand means the data we are operating upon (in most cases source data). It can be a direct address of memory, it can be register names, it can be any numerical data , Explain this with a simple data move instruction of 4545 below:

MOV A,30h

Answer

The data A is the operand, often known as source data. When this instruction is executed, the data 30h is moved to accumulator A. There are five different ways to execute this instruction and hence we say, we have got five addressing modes for 4545. Identify and illustrate these seven addressing modes types as explained above.

3. An array of two integers (each integer = 32 bits) is placed in memory starting with address 100. Show how to increment each element of the array using register indirect addressing mode.

Answer

4. LOAD r1, 100# the base
5. LOAD r2, 1 # 1 will be used for increment
6. ADD r3, r2, (r1)
7. STORE (r1), r3
8. ADD r1, r1, 4# the next array element is at 104
9. ADD r3, r2, (r1)
10. STORE (r1), r3

Unit Readings and Other Resources

The readings in this unit are to be found at course level readings and other resources.

1. www.cs.virginia.edu/~cs333/notes/cs333_class3.pdf- class 3 Instruction set Architectures, University of Virginia

2. http://www.slideshare.net/srisumandas/2instruction-set-architecture-design- Instruction set Architecture

3. http://www.encyclopedia.com/doc/1O11-instructionformat.html JOHN DAINTITH. "instruction format." A Dictionary of Computing. 2004. Encyclopedia.com. 12 Mar. 2016 <http://www.encyclopedia.com>.

4. http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Mips/format.html Instruction format

5. http://en.wikipedia.org/wiki/MIPS_instruction_set MIPS instruction set

6. http://en.wikipedia.org/wiki/Computer_architecture computer architecture

7. http://simple.wikipedia.org/wiki/Computer_architecture - computer architecture

8. http://en.wikiversity.org/wiki/Computer_architecture_and_organization - Computer architecture and organization. http://en.wikipedia.org/wiki/Computer_architecture-

9. http://www.karbosguide.com/books/pcarchitecture/start.htm PC Architecture a book by Michael Karbo. chapter 29

10. http://en.wikipedia.org/wiki/Multiprocessing multiprocessing

11. http://www.personal.kent.edu/~aguercio/CS35101Slides/Tanenbaum/CA_Ch05_PartII.pdf - common Instruction Formats.

12. https://en.wikibooks.org/wiki/A-level_Computing/AQA/Computer_Components,_The_Stored Program_Concept_and_the_Internet/Machine_Level_Architecture/Machine_code_and_processor_instruction_set_Machine_Level_Architecture/Machine_code_and_processor_instruction_set

13. http://homepage.cs.uri.edu/book/cpu_memory/http://cs.sru.edu/~mullins/cpsc100book/IntroToCS.htmlcpu_memory.htm How Computers Work: The CPU and Memory

14. http://www.tutorialspoint.com/computer_fundamentals/index.htm- Computer Fundamentals

15. http://en.wikipedia.org/wiki/Computer_architecture#Instruction_set_architecture computer architecture

16. http://www.cs.toronto.edu/~demke/469F.06/Lectures/Lecture6.pdf- Lecture on Interrupts (University of Toronto)

17. http://people.freebsd.org/~jhb/papers/bsdcan/2007/article/article.html PCI Interrupts for x86 Machines under FreeBSD

18. http://en.wikipedia.org/wiki/Instruction_set Instruction set

Course Summary

Computer organization and architecture is about the components of the computer system, how they interconnect and how they function. You should therefore be able to identity the main component of the computer systems and the how these components are interconnected. The basic block of a functional computer components are logic circuits. The course has made you to acquire knowledge on how to construct different logic circuits given any inputs and corresponding expected outputs.

The data representation in a computer is in terms of binary systems and it is expected that you can now evaluate different data representation. You have learnt different addressing modes which the computer uses to process the data. The module is a prerequisite to advanced computer organization and architecture.

Course Assessment

There will be a final examination covering all the units of this course. It will have 70% of the total marks for the course, The grading is as shown in the grading scheme.

Grading Scheme

Three unit assignment of varying complexity and weights. These are the ones included at the end of each of the units 1 to 4.	20%
Mid Term exam	10%
Final exam	70%

Course Assessment

[Final Exam]

Time:- 3 hours-

It is advisable to spend approximately 3.6 minutes per question.

This is the final examination for this module. It will examine all the units covered for the course.The exam will carry 70% of the marks for this course. This will be added to 20% covered (done) as part of the assessments for the respective units and 10% for the MID term continuous assessment test to make 100% as the total for the course.

Instructions

You are to attempt ALL questions and your answers must be in the format detailed by each question. Each question has marks to be awarded on successful completion. There are four parts, each representing the unit covered in the course (module).

PART A- System Architecture and Design [18 marks]

a. Name three major functional units of a computer. (1 mk for each =3 marks)

b. Briefly explain the main functions of each of these components. (1 mk for each = 3 marks)

c. Which unit are the following devices associated with

i. Keyboard

ii. Mouse

iii. Screen

iv. Joystick

v. Webcam

vi.Speaker (0.5 mks each = 3 marks)

d. Which is the most widely used input device when entering text? (0.5 marks)

e. Name the part of the computer where all other components connect to. (0.5 marks)

f. What is the main use of the computer's main memory? (2 marks)

g. The computer's main memory is of two types. List and explain how these types are used by the computer. (10 marks)

PART B- Data representation and logic [31 marks]

a. What do you understand by a number system? (1 mark)

b. Briefly describe the characteristics of the following number systems

i. Decimal

ii. Hexadecimal

iii. Binary (@1 mk = 3 marks)

c. Convert the following numbers into the number system given in the question. Show how you have arrived at the answer.

i. 111022 to base 10
(4 marks)

ii. 2110 to binary (4 marks)

d.What do you understand by a logic gate? (2 marks)

e. What is a truth table when used in electronics? (2 marks)

f. Describe the basic operations of the following logic gates. Include corresponding truth tables for each logic gate. The diagram representing each gate must be given. Presuppose that there are two inputs variables, A and B and one output C for each.

i. AND (2 marks)
ii. OR (2 marks)

iii. NOT (2 marks)

iv. NAND (2 marks)

v. NOR (2 marks)

PART C- Computer Organization [11 marks]

a) What do you understand by the following terms. Include a diagram to illustrate this.

i) Control bus

ii) Address bus

iii) Data bus (3 marks)

b) Differentiate between the following terms

i) Memory address register

ii) Memory data register (4 marks)

c) Name three important steps that are followed by a computer in executing an instruction. (1 mk each = 3 marks)

d)What is the name of the register that stores the address of the next instruction to be fetched from memory? (2 marks,)

e)What is the name of the register that is used to store an instruction that is awaiting decoding or execution? (2 marks,)

PART D- Instruction Set Architecture [10 marks]

a) What does ISA stand for? (1 mark)

b) What do you understand by

i) a memory address? (1 mark)

ii)Addressing mode? (1 marks)

c) Explain the mode of operation of each the following addressing modes. Use of a diagram to support your explanation.

i) Direct (absolute) addressing

ii) Indirect addressing

iii) Relative addressing

iv) Immediate addressing (@1 mark= 4 marks)

d)Explain what you understand by the following instructions

i) MOV A, R2 (1 marks)

ii) MOV A, @R0 (1 marks)

iii) MOV @R1,A (1 marks)

EXAM

SAMPLE ANSWERS

PART A- System Architecture and Design

a. Name three major functional units of a computer.

I expect the following as the units:-

- Input
- Output
- Central processing unit

b.Briefly explain the main functions of each of these components.

Full marks for correct functional descriptions, 0 otherwise

c. Which unit are the following devices associated with

i. Keyboard--Input unit

ii. Mouse- input unit

iii. Screen- output unit

iv. Joystick- input unit

v. Webcam-

vi. Speaker- output unit

d. Which is the most widely used input device when entering text? It is the keyboard
e. Name the part of the computer where all other components connect to. This is the motherboard

PART B- Data representation and logic [ ]

a. What do you understand by a number system?- A Number system is a set of values used to represent different quantities.

b. Briefly describe the characteristics of the following number systems

i. Decimal- This consists of ten digits from 0 to 9. These digits can be used to represent any numeric value. The base of decimal number system is 10 as it uses ten digits.

ii. Hexadecimal-- is consist of 16 digits from 0 to 9 and A to F. The alphabets A to F represent decimal numbers from 10 to 15. The base of this number system is 16.

iii. Binary- 'Binary' means there are only 2 possible values: 0 and 1. One binary digit (0 or 1) is referred to as a bit, which is short for binary digit.

c. Convert the following numbers into the number system given in the question. Show how you have arrived at the answer.

i. 111022 to base 10 - ((1 * 24) + (1 * 23) + (1 * 22) + (0 * 21) + (1 * 20))10 =(16 +8+4+0+1)=2910 (4 marks for correct methodology, 0 otherwise)

ii. 2110 to binary= 1011012 I expect the answer to include successive division by 2 and recording the remainder until correct answer is found.(3 marks for correct method, 0 otherwise)

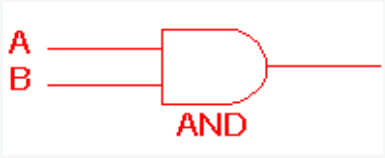
d. What do you understand by a logic gate? This is an elementary building block of a digital circuit.

e. What is a truth table when used in electronics? It is a diagram of the outputs from all possible combinations of all inputs.

f. Describe the basic operations of the following logic gates. Include corresponding truth tables for each logic gate. The diagram representing each gate must be given. Presuppose that there are two inputs variables, A and B and one output C for each.

The following will be expected for each as the answer; logic circuit diagram (1 mark) + (.5 marks) truth table and brief description (.5 marks)

i. AND



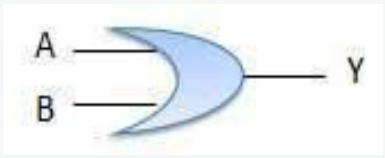
This gate got its name because of the logical "AND" operator that requires that the values of the participating variables be either both true or both false. The output will be if both inputs are TRUE else it will be "false".

The associated truth table would be like

INPUT		OUTPUT
A	B	AB =C
0	0	0
0	1	0
1	0	0
1	1	1

ii. OR

The diagram would be like



where Y=C

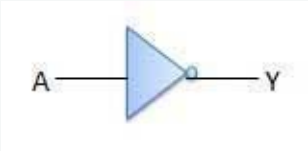
The output will be 1 (TRUE) if either or both the inputs are "TRUE". If both inputs are "false," then the output is "false." The corresponding truth table would be like

Inputs outputs

Inputs		outputs
A	B	A+B=C
0	0	0
0	1	1
1	0	1
1	1	1

iii. NOT

This is a circuit that produces an inverted version of its input. Its logic diagram is like. THis is for a single input A and output Y example



The truth table would be like

INPUT	OUTPUT
A	Y (NOT A)
0	1
1	0

iv. NAND

This kind of gate works like the AND gate only that the AND gate must be followed by a NOT gate. The diagram would be like

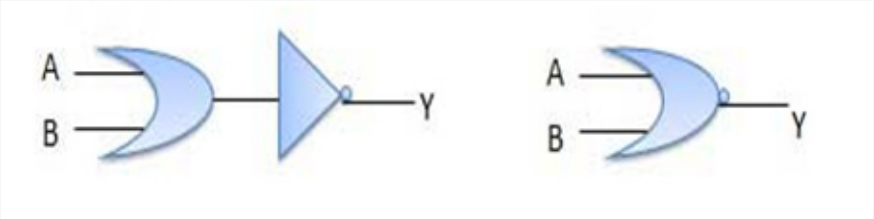
The truth table would be

Inputs Output

Inputs		Output
A	B	NOT(A and B)
0	0	1
0	1	1
1	0	1
1	1	0

v. NOR

The NOR gate is a combination OR gate followed by an inverter.



Inputs		Output
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

PART C- Computer Organization

a. What do you understand by the following terms- I expect the following,; {full marks, 0 otherwise}

i) Control bus- A control bus is a computer bus that is used by the CPU to communicate with devices that are contained within the computer. This occurs through physical connections such as cables or printed circuits.

ii) Address bus- An address bus is a computer bus architecture used to transfer data between devices that are identified by the hardware address of the physical memory (the physical address), which is stored in the form of binary numbers to enable the data bus to access memory storage.

iii) Data bus- A data bus is a system within a computer or device, consisting of a connector or set of wires, that provides transportation for data

b) Differentiate between the following terms

i)Memory address register- is a CPU register that either stores the memory address from which data will be fetched to the CPU or the address to which data will be sent and stored

ii) Memory data register- is the register of a computer's control unit that contains the data to be stored in the computer storage (e.g. RAM), or the data after a fetch from the computer

c) Name three important steps that are followed by a computer in executing an instruction. I expect the mention of fetch, decode and execute

d) What is the name of the register that stores the address of the next instruction to be fetched from memory? It is program counter/ PC

e) What is the name of the register that is used to store an instruction that is awaiting decoding or execution? It is instruction register/ IR

PART D- Instruction Set Architecture

a) What does ISA stand for?- It short for Instruction Set Architecture

b) What do you understand by the following terms?

i) a memory address?- It is a unique identifier (address) often assigned at boot time that is used by the CPU to track data in the system.

ii) Addressing mode?- this is a way of specifying the location (address) of an operand in an instruction.

c) Differentiate between the following addressing modes

i) Direct (absolute) addressing

ii) Indirect addressing

iii) Relative addressing

iv) Immediate addressing

d) Explain what you understand by the following instructions

i) MOV A, R2, It means copy R2 into A

ii) MOV A, @R0- It means move the contents of RAM location whose address is held by R0 into A.

iii)MOV @R1,A - This means move the contents of A into RAM location whose address held by R1.

Course Reference

1. http://www.karbosguide.com/books/pcarchitecture/start.htm- PC Architecture a book by Michael Karbo- chapter 2 to chapter 8.

2. http://simple.wikipedia.org/wiki/Computer_architecture - computer architecture

3. http://en.wikiversity.org/wiki/Computer_architecture_and_organization - Computer architecture and organization.

4. http://www.tutorialspoint.com/computer_fundamentals/index.htm- Computer Fundamentals- (Read from computer types to computer hardware)

5. http://en.wikipedia.org/wiki/Computer_architecture#Instruction_set_architecture- computer architecture.

6. http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Mips/format.html Instruction format

7. http://en.wikipedia.org/wiki/MIPS_instruction_set MIPS instruction set

8. http://en.wikipedia.org/wiki/Computer_architecture computer architecture

9. http://simple.wikipedia.org/wiki/Computer_architecture - computer architecture

10. http://en.wikiversity.org/wiki/Computer_architecture_and_organization - Computer architecture and organization.http://en.wikipedia.org/wiki/Computer_architecture.

11. http://www.karbosguide.com/books/pcarchitecture/start.htm PC Architecture a book by Michael Karbo. chapter 29.

12. http://en.wikipedia.org/wiki/Multiprocessing multiprocessing.

13. http://homepage.cs.uri.edu/book/cpu_memory/http://cs.sru.edu/~mullins/cpsc100book/IntroToCS.htmlcpu_memory.htm How Computers Work: The CPU and Memory.

14. http://www.tutorialspoint.com/computer_fundamentals/index.htm- Computer Fundamentals

15. http://en.wikipedia.org/wiki/Computer_architecture#Instruction_set_architecture computer architecture.

16 http://www.cs.toronto.edu/~demke/469F.06/Lectures/Lecture6.pdf- Lecture on Interrupts (University of Toronto)

17. http://people.freebsd.org/~jhb/papers/bsdcan/2007/article/article.html PCI Interrupts for x86 Machines under FreeBSD

18. http://en.wikipedia.org/wiki/Instruction_set Instruction set

The African Virtual University Headquarters

Cape Office Park

Ring Road Kilimani

PO Box 25405-00603

Nairobi, Kenya

Tel: +254 20 25283333

contact@avu.org

oer@avu.org

The African Virtual University Regional Office in Dakar

Université Virtuelle Africaine

Bureau Régional de l'Afrique de l'Ouest

Sicap Liberté VI Extension

Villa No.8 VDN

B.P. 50609 Dakar, Sénégal

Tel: +221 338670324

bureauregional@avu.org